JBoss Enterprise Application Platform 5
Release Notes 5.1.0

---

zum Gebrauch mit der JBoss Enterprise Application Platform 5.1.0
Ausgabe 5.1.0

Laura Bailey Rebecca Newton Joshua Wulf

JBoss Enterprise Application Platform 5 Release Notes 5.1.0

---

zum Gebrauch mit der JBoss Enterprise Application Platform 5.1.0
Ausgabe 5.1.0

Laura Bailey
lbailey@redhat.com

Rebecca Newton
rnewton@redhat.com

Joshua Wulf
jwulf@redhat.com

Rechtlicher Hinweis

Copyright © 2011 Red Hat.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Zusammenfassung

Diese Release Notes enthalten wichtige Informationen im Zusammenhang mit der JBoss Enterprise Application Platform 5.1.0, die noch nicht in den Produkthandbüchern enthalten sind. Sie sollten diese Release Notes vollständig lesen, bevor Sie die JBoss Enterprise Application Platform 5.1.0 installieren.

Inhaltsverzeichnis

1. Einführung **3**
1.1. Übersicht 3

2. Neue Features in der JBoss Enterprise Application Platform 5.1.0 . . . . . **3**
2.1. Red Hat Enterprise Linux 6 Support 3
2.2. Das Native-Paket 4
2.3. JBoss Web Services 4
2.4. Hibernate 4
2.5. PicketLink Federation 4
2.6. In vorherigen Releases präsentierte Features 4
2.6.1. Neue Features in der JBoss Enterprise Application Platform 5.0.1 4
2.6.1.1. Das Native-Paket 4
2.6.1.1.1. Native Solaris SPARC Pakete 5
2.6.2. Neue Features in der JBoss Enterprise Application Platform 5.0 5
2.6.2.1. JBoss AS 5
2.6.2.2. JBoss Microcontainer 5
2.6.2.3. JBoss Messaging 5
2.6.2.4. JBoss Cache 5
2.6.2.5. JBoss Web Services Native 5
2.6.2.6. Das Native-Paket 5

3. Komponentenversionen **6**

4. Installationshinweise **8**
4.1. Unterstützte Konfigurationen 8
4.2. Installation der JBoss Enterprise Application Platform 8
4.3. Standardmäßiges Startprofil 8
4.4. Quelldateien 8

5. Produkt-Support **8**

6. Dokumentation **8**

7. Technologievorschauen **9**
7.1. PicketLink 9

8. Ausgenommene, entfernte und veraltete Posten **9**
8.1. Ausgeschlossene Posten 10
8.2. Veraltete Posten 10
8.3. Entfernte Elemente 10

9. In dieser Release behobene Probleme **10**
9.1. Allgemeine, behobene Probleme 10
9.2. Behobene Probleme mit Hibernate 12
9.3. Behobene Remoting Probleme 15
9.4. Behobene Probleme mit Seam 15
9.5. Behobene Security Probleme 19
9.6. Behobene Applikationsserver Probleme 19
9.7. Behobene Dokumentationsprobleme 19
9.8. Behobene Probleme bei Messaging 20
9.9. Behobene Probleme mit Web Services 20
9.10. Behobene Probleme mit RESTEasy 21
9.11. Behobene Clustering Probleme 21
9.12. Behobene Cache Probleme 21

9.13. Behobene Connector Probleme 22

10. Bei dieser Release bekannte Probleme 22
10.1. Nicht unterstützte JARs 22
10.2. Allgemeine, bekannte Probleme 22
10.3. Bekannte Installer-Probleme 26
10.4. Bekannte Probleme mit Hibernate 26
10.5. Bekannte Probleme mit JBoss Messaging 33
10.6. Bekannte Probleme mit JBoss Web Services 34
10.7. Bekannte Probleme mit Remoting 35
10.8. Bekannte Probleme mit RESTEasy 35
10.9. Bekannte Probleme mit Seam 35
10.10. Bekannte Probleme mit EJB3 36
10.11. Bekannte Probleme mit Clustering 37
10.12. Bekannte Probleme mit dem Connector 37

A. Änderungsverzeichnis 39

1. Einführung

Diese Release Notes enthalten wichtige Informationen zur JBoss Enterprise Application Platform 5.1.0. Neue Features, in dieser Release behobene Probleme sowie andere aktuelle Themen werden hier behandelt.

1.1. Übersicht

Über die JBoss Enterprise Application Platform

Die JBoss Enterprise Application Platform ist der nächste Evolutionsschritt in Open Source Enterprise Software. Sie ist ein leistungsfähiges Tool zur Entwicklung umfassender, sehr leistungsfähiger Web 2.0 Anwendungen auf einer reinen Java-Plattform.

Durch Integration der besten Open-Source-Frameworks ihrer Art, wie etwa JBoss Seam, Hibernate, CXF Web Services, JBoss Cache und JBoss Messaging nutzt die Plattform die Innovationen der Open Source Community. Die JBoss Enterprise Application Platform ist vollständig getestet und wird von Red Hat unterstützt. Desweiteren ist sie für den Einsatz mit vielen führenden Hardware- und Software-Produkten der Unternehmensklasse zertifiziert.

Zu dieser Release

JBoss Enterprise Application Platform 5.1.0 ist eine *kleinere Release*. Kleinere Releases verbinden die Inhalte vorheriger Patches und Cumulative Patches (CPs) und können zusätzliche neue Funktionalität hinzufügen. Nachfolgende Patches und Cumulative Patches gehen davon aus, dass die Installation des kleineren Updates ihnen vorausgegangen ist. Während des Lebenszyklus der Hauptrelease eines Produkts bemüht sich Red Hat in kommerziell sinnvollem Rahmen Kompatibilität auf API-Ebene über alle kleineren Releases und asynchronen Patches hinweg zu erhalten, so dass zum Beispiel sichergestellt ist, dass die JBoss Enterprise Application Platform 5.1.0 Kompatibilität auf API-Ebene mit der JBoss Enterprise Application Platform 5.0.0, der ursprünglichen Release der JBoss Enterprise Application Platform 5 bewahrt. Mögliche Ausnahmen zu dieser Regel sind Fehlerbehebungen an kritischen Sicherheitsproblemen.

Mit der Release der JBoss Enterprise Application Platform 5.1.0 sollten JBoss Enterprise Application Platform 5.0.x Kunden eine Aktualisierung zur JBoss Enterprise Application Platform 5.1.0 durchführen.

Weitere Informationen finden Sie in der JBoss Enterprise Middleware Product Update and Support Policy.

2. Neue Features in der JBoss Enterprise Application Platform 5.1.0

JBoss Enterprise Application Platform 5.0.x enthielt viele aktualisierte und neue Komponenten. Die JBoss Enterprise Application Platform 5.1.0 fügt Performance Verbesserungen, das JBoss Web Services CXF Stack und Support für JDBC 4 hinzu und stellt PicketLink Federation als Technologievorschau vor. Diese Komponenten werden in diesem Abschnitt erläutert.

2.1. Red Hat Enterprise Linux 6 Support

Die ZIP-Distribution der JBoss Enterprise Application Platform 5.1 ist vollständig für den Gebrauch mit Red Hat Enterprise Linux 6 getestet. Die entsprechenden Native Pakete sind unter http://access.redhat.com verfügbar.

2.2. Das Native-Paket

Das Native Package gemäß folgendem Muster **`jboss-ep-native-5.1.0*.zip`** umbenannt. Dieses neue, konsolidierte Native Paket kann mit der JBoss Enterprise Application Platform oder mit der JBoss Enterprise Web Platform verwendet werden.

Das Native Paket wurde außerdem erweitert, um NSAPI für die Solaris Technologie zu liefern. NSAPI ist ein Connector der dazu verwendet wird, den Sun Java System Web Server mit der JBoss Enterprise Application Platform zu verbinden.

2.3. JBoss Web Services

JBoss Enterprise Application Platform 5.0 bietet eine einbindbare Architektur um eine Auswahl von Web Services Stack mit JBoss Web Services Native als dem unterstützten Web Services Provider zu liefern. JBoss Enterprise Platform 5.1.0 präsentiert ein alternatives Web Services Stack namens JBoss Web Services CXF.

Anmerkung

Die Wahl des von der Plattform verwendeten Web Services Stacks muss zum Zeitpunkt der Installation erfolgen. Weitere Informationen finden Sie im *Installation Guide*.

JBoss Web Services Native

JBoss Web Services Native ist die Java EE 5-konforme JBoss Implementierung von Web Services Standards. Es ist das einzige Web Services Stack für Versionen von JBoss Enterprise Application Platform, die älter als 5.1 sind und es ist das standardmäßige Web Services Stack in der JBoss Enterprise Application Platform 5.1.0.

JBoss Web Services CXF

JBoss Web Services CXF bietet die meisten in Apache CXF (einschließlich WS-Security, WS-Policy, WS-Addressing, WS-ReliableMessaging, grundlegendes WS-Trust, MTOM) verfügbaren Features sowie gängige JBoss Web Services Stack Features wie Endpoint-Metrics, Record Management und Endpoint Address Rewrite. JBoss Enterprise Application Platform 5.1.0 präsentiert JBoss Web Services CXF Stack als ein optionales Web Services Stack.

2.4. Hibernate

Hibernate bietet jetzt kompletten Support für Java Database Connectivity 4 (JDBC 4), welches besseren Clob und Blob Support und eine verbesserte Handhabung von Ausnahmen bietet.

2.5. PicketLink Federation

PicketLink Federation ist ein Security und Identity Framework das Support für SAML v2.0 und WS-Trust liefert. Die JBoss Enterprise Application Platform 5.1.0 präsentiert PicketLink als eine Technologievorschau.

2.6. In vorherigen Releases präsentierte Features

2.6.1. Neue Features in der JBoss Enterprise Application Platform 5.0.1

2.6.1.1. Das Native-Paket

Das Native Paket wurde erweitert, um mod_cluster und ISAPI für Windows Technologien zusätzlich zu

JBoss Native und mod_jk zu liefern. Diese hinzugefügten Technologien lassen sich folgendermaßen beschreiben:

- **mod_cluster** ist ein httpd-basierter Lastverteiler. Wie **mod_jk** verwendet es einen Kommunikationskanal zur Weiterleitung von Anfragen von httpd an einen Applikationsserverknoten; es verfügt jedoch auch über einen Kommunikationskanal von Serverknoten zum Lastverteiler, wodurch Feedback in Echtzeit geliefert und effizientere Lastoptimierung und Anfragen-Routing ermöglicht werden. In der mod_cluster Dokumentation unter http://docs.redhat.com/ finden Sie weitere Informationen dazu.
- **ISAPI** ist ein Connector, der zur Verbindung des Microsoft IIS Web Servers mit der JBoss Enterprise Application Platform verwendet wird.

2.6.1.1.1. Native Solaris SPARC Pakete

Native Solaris SPARC Pakete werden ab dieser Release von JBoss Enterprise Application Platform bereitgestellt. Im JBoss Enterprise Application Platform *Installation Guide* finden Sie Installationsanweisungen für native Pakete.

2.6.2. Neue Features in der JBoss Enterprise Application Platform 5.0

2.6.2.1. JBoss AS

Die JBoss AS 5 Familie ist die hochmoderne, auf Microcontainern der zweiten Generation basierende Enterprise Java Run-Time und die neueste Version des weltweit beliebesten Applikationsservers. Er unterstützt nicht nur die aktuellste Java EE Spezifikation (Java EE 5), sondern integriert darüberhinaus viele der besten Dienste der Unternehmensklasse für erweitertes Messaging, Persistenz, Transaktionen, Caching und Hochverfügbarkeit.

2.6.2.2. JBoss Microcontainer

Der JBoss Microcontainer ist eine Umgestaltung des modularen JBoss JMX Microkernels. Es handelt sich dabei um einen schlanken Kernel, der das Laden, den Lebenszyklus sowie Abhängigkeiten zwischen POJOs verwaltet. Der JBoss Microcontainer wird zur Integration von Unternehmensdiensten eingesetzt und bietet zusammen mit Servlet/JSP-Container, EJB-Container, Deployern und Verwaltungsprogrammen ein standardmäßiges Java EE 5 Profil.

2.6.2.3. JBoss Messaging

JBoss Messaging ist der standardmäßige JMS Provider der JBoss Enterprise Application Platform 5.1. Es handelt sich um einen kompletten Rewrite von JBossMQ, dem bestehenden JBoss JMS Provider, der verglichen mit JBossMQ die Performance in sowohl Einzel-Node als auch geclusterten Umgebungen verbessert.

2.6.2.4. JBoss Cache

JBoss Cache wird zur Replikation von EJB- und HTTP-Session-Zuständen verwendet und unterstützt verteiltes Entity-Caching für JPA, während mit einem neuen, effizienteren Sperrschema (MVCC – MultiVersion Concurrency Control) weiterhin die Leistung und Skalierbarkeit bestmöglich ausgereizt wird.

2.6.2.5. JBoss Web Services Native

JBoss Web Services Native is t die Java EE 5-konforme JBoss Implementierung von Web Services Standards. Es ist das einzige Web Services Stack für Versionen der JBoss Enterprise Application Platform vor 5.1, und es ist das standardmäßige Web Services Stack in JBoss Enterprise Application Platform 5.1.

2.6.2.6. Das Native-Paket

Das Native-Paket ist eine optionale Komponente für die JBoss Enterprise Application Platform und umfasst JBoss Native und die mod_jk-Technologien. Diese Technologien lassen sich folgendermaßen beschreiben:

- JBoss Native umfasst Apache Portable Runtime (APR), OpenSSL und Tomcat Native (TC-native).
  - **Apache Portable Runtime (APR)** bietet ausgezeichnete Skalierbarkeit und Leistung, sowie verbesserte Integration mit nativen Server-Technologien. Die Apache Portable Runtime ist eine hochportable Bibliothek, die das Herzstück des Apache HTTP Server 2.x bildet und viele Einsatzmöglichkeiten bietet, u.a. Zugriff auf erweiterte IO-Funktionalitäten (z.B.: sendfile, epoll und OpenSSL), Funktionalitäten auf Betriebssystemebene (z.B.: Erzeugung zufälliger Zahlen und Systemstatus), sowie Handhabung nativer Prozesse (gemeinsam verwendeter Speicher, NT-Pipes und Unix-Sockets).
  - **OpenSSL** implementiert die Secure Sockets Layer (SSL) und Transport Layer Security (TLS) Protokolle und enthält eine einfache krytografische Bibliothek.
  - **Tomcat Native (TC-Native)** ist ein Java Native Interface (JNI), das den Großteil der Kernfunktionalität von Tomcat in nativem Code anstelle von Java bereitstellt. Dies ermöglicht einen insgesamt schnelleren Server.
- **mod_jk** ist ein Connector, der zur Anbindung des Tomcat JSP Containers an verschiedene Webservices wie Apache verwendet wird.

Anmerkung

Die Red Hat Enterprise Linux 5 Distribution des Native-Pakets umfasst nicht OpenSSL oder die Apache Portable Runtime, da diese Technologien bereits im Betriebssystem enthalten sind.

3. Komponentenversionen

Dieser Abschnitt führt die Versionen der einzelnen Komponenten auf, aus denen sich die Enterprise Application Platform 5.1 zusammensetzt.

Tabelle 1. Vergleich der Komponentenversion zwischen Releases 5.1.0 und 5.0.1

Komponente	EAP 5.1	EAP 5.0.1
JBoss Application Server	5.1.0.GA	5.1.0.GA
JBoss Microcontainer	2.0.10.GA	2.0.9.GA
JBoss Native	2.0.9.GA	2.0.9.GA
Hibernate Core	3.3.2.GA_CP03	3.3.2.GA_CP01
Hibernate Entity Manager	3.4.0.GA_CP03	3.4.0.GA_CP01
Hibernate Annotationen	3.4.0.GA_CP03	3.4.0.GA_CP01
Hibernate Search	3.1.1.GA_CP03	3.1.1.GA_CP01
Hibernate Validator	3.1.0.GA	3.1.0.GA
Geclusterte Remote-EJB3-Proxys	1.3.6.GA	1.1.22.GA
JBoss Cache	3.2.5.GA	3.2.1.GA
JBoss HA Server API	1.1.3.Final	1.1.1.GA
JBoss JAXR	2.0.1	2.0.1
JGroups	2.6.16.GA	2.6.13.GA
JBoss EJB3	1.3.6.GA	1.1.22.GA
JBoss JTA	4.6.1.GA_CP07	4.6.1.GA_CP03
JBoss JTS	4.6.1.GA_CP07	4.6.1.GA_CP03
JBoss Negotiation	2.0.3.SP3	2.0.3.SP1
JBoss Managed	2.1.1.GA	2.1.1.GA
JBoss Messaging	1.4.7.GA	1.4.6.GA
JBoss Metadata	1.0.6.GA	1.0.2
JBoss Web	2.1.10.GA	2.1.7.GA
JBoss Web Services - Native	3.1.2.SP7	3.1.2.SP3
JBoss Web Services - CXF	3.1.2.SP7	-
JBoss AOP	2.1.6.CP01	2.1.6.GA
JBoss Remoting	2.5.3.SP1	2.5.2
JBoss Serialization	1.0.4.FINAL	1.0.3.GA
JBoss XB	2.0.1.GA	2.0.1.GA
JavaServer Faces	1.2_13	1.2_13
JacORB	2.3.1.jboss.patch01	2.3.1.jboss.patch01
JPA	1.0.0	1.0.0
JBoss Security	2.0.4.SP4	2.0.4.SP1
JBoss Profiler-jvmti	1.0.0.CR5	1.0.0.CR5
Admin-Konsole	1.3.4.SP1	1.3.2.GA
mod_jk	1.2.30	1.2.28
mod_cluster	1.0.4.GA	1.0.3.GA
Seam	2.2.2.EAP5	2.2.1.EAP5
RESTEasy	1.2.GA_CP01	1.1.GA_CP01
RichFaces (in Seam)	3.3.1.SP2	3.3.1.GA

Technologievorschau

JBoss Enterprise Application Platform 5.1 enthält PicketLink Federation als Technologievorschau. Technologievorschau-Features werden im Rahmen der Red Hat Subscription Level Agreements (SLAs) nicht vollständig unterstützt, sind in Ihrer Funktion ggf. eingeschränkt und allgemein nicht für den Produktionsbetrieb geeignet. Diese Features bieten jedoch frühzeitigen Zugriff auf zukünftige Produktinnovationen, so dass Kunden die Funktionalität testen und noch während des Entwicklungsprozesses wertvolles Feedback geben können. Da Red Hat beabsichtigt, zukünftige Versionen der Technologievorschau-Features für die Allgemeinheit zu veröffentlichen, werden wir uns in wirtschaftlich vertretbarem Umfang bemühen, sämtliche Fehler zu beseitigen, die von Kunden beim Verwenden dieser Features beobachtet werden.

4. Installationshinweise

4.1. Unterstützte Konfigurationen

Eine aktuelle Matrix kompatibler und zertifizierter Konfigurationen ist unter http://www.jboss.com/products/platforms/application/supportedconfigurations/ verfügbar. In dieser Liste finden Sie Informationen zu getesteten und unterstützten Konfigurationen.

4.2. Installation der JBoss Enterprise Application Platform

Das **`Installationshandbuch`** erläutert, wie die JBoss Enterprise Application Platform installiert und verifiziert wird.

4.3. Standardmäßiges Startprofil

Das standardmäßige Startprofil ist **`default`**, ein einfaches Java EE 5 Server-Profil mit einer Reihe von Standarddiensten. Es umfasst die am häufigsten verwendeten Dienste, die zum Implementieren einer Java EE 5 Applikation nötig sind. Es enthält weder den JAXR-Dienst noch den IIOP-Dienst oder jegliche Clustering-Dienste.

Das **`default`**-Profil ist nicht für den Einsatz in Produktionsumgebungen oder für die Durchführung von Last-, Stress-, Verfügbarkeits- oder Leistungstests ausgelegt. Außerdem repräsentiert die Performance dieser Beta-Release nicht notwendigerweise die der endgültigen Release.

4.4. Quelldateien

Quell-ZIP

- ftp://ftp.redhat.com/pub/redhat/jbeap/5.1.0/en/source/jboss-eap-src-5.1.0.zip

5. Produkt-Support

Fehler, potenzielle Fehler und Entwicklungsprobleme und/oder -fragen sind ein Fall für den JBoss-Support und sollten über das JBoss Customer Support Portal unter http://access.redhat.com gemeldet werden.

6. Dokumentation

In dieser Release wird die JBoss Transactions Dokumentation in den *Transactions Development Quick*

Start Guide, den *Transactions Administrators Guide* und den *Transactions Development Guide* restrukturiert und konsolidiert.

In dieser Release ist außerdem der *JBoss Security Guide* hinzugefügt, der die Sicherheitsinformationen der Plattform enthält.

JBoss WS CXF wird ebenfalls in dieser Release eingeführt und enthält ein Benutzerhandbuch zum Dokumentationssatz der Plattform.

Alle API-Dokumente, Code-Beispiele und ein Link zu den online Release Notes werden in einer separaten ZIP-Datei distribuiert **`jboss-eap-docs-<version>.zip`**.

Alle anderen Handbücher und Dokumentationen sind online unter http://docs.redhat.com verfügbar. Wählen Sie den jeweiligen Produktnamen (JBoss Enterprise Application Platform) und Stream (5) aus der Seitenleiste wie in der Abbildung auf der Vorderseite dargestellt.

Wir freuen uns über Ihr Feedback zu unserer Dokumentation. Sie können das Dokumentationsteam über http://jira.jboss.org direkt kontaktieren.

7. Technologievorschauen

Dieser Abschnitt beschreibt die Technologievorschau-Features, die zusammen mit der JBoss Enterprise Application Platform veröffentlicht werden, deren Installation, sowie jegliche bekannte Probleme.

Technologievorschau

Technologievorschau-Features werden im Rahmen der Red Hat Subscription Level Agreements (SLAs) nicht vollständig unterstützt, sind in Ihrer Funktion ggf. eingeschränkt und allgemein nicht für den Produktionsbetrieb geeignet. Diese Features bieten jedoch frühzeitigen Zugriff auf zukünftige Produktinnovationen, so dass Kunden die Funktionalität testen und noch während des Entwicklungsprozesses wertvolles Feedback geben können. Da Red Hat beabsichtigt, zukünftige Versionen der Technologievorschau-Features für die Allgemeinheit zu veröffentlichen, werden wir uns in wirtschaftlich vertretbarem Umfang bemühen, sämtliche Fehler zu beseitigen, die von Kunden beim Verwenden dieser Features beobachtet werden.

7.1. PicketLink

PicketLink Federation liefert der Plattform Identity Federation und Single Sign-on mit Support für SAML 2.0, WS-Trust 1.3, XACML 2.0 (via JBossXACML) und OpenID 1.1 und 2.0. Im *Installation Guide* finden Sie Informationen zur Installation von PicketLink.

8. Ausgenommene, entfernte und veraltete Posten

Ausgenommene, veraltete und entfernte Posten

Ausgenommen

Ein Posten, der nie Teil einer Produkt-Release war, jedoch sonst Teil einer der Open Source Komponenten des Produkts ist, ist ein *excluded item* (ausgeschlossener Posten).

Veraltet

Ein Posten, der aus einer zukünftigen Release (in der Regel der nächsten Hauptversion) entfernt wird, gilt als *deprecated item* (veralteter Posten).

Entfernt

Ein Posten, der in der Vergangenheit Teil einer Release des Produkts war und nicht länger enthalten ist, ist ein *removed item.* (entfernter Posten). Posten gelten in der Regel als veraltet, ehe sie entfernt werden.

8.1. Ausgeschlossene Posten

In dieser Release sind keine Posten ausgenommen.

8.2. Veraltete Posten

Die nachfolgenden Posten gelten in dieser Release als veraltet:

- JBoss Cache JBAS-7837
  - Pojo Cache FIELD Replikationsgranularität JBAS-7560
- JMX Console JBAS-7509
  - JFreeChart JBAS-7716
- JMX Remoting JBAS-7510
- org.jboss.jdo.castor.CastorJDOImpl MBean JBAS-7511
- JDBC3 (JDK5) Klassen im JDBC-Adapter JBAS-7527
- Java Remote Method Protocol Invoker und HA Invoker JBAS-7563
- PooledInvoker und PooledInvokerHA JBAS-7564
- HASessionStateTopologyComputer und HASessionStateTopologyComputerImpl JBAS-7992
- SubPartitionsInfo und SubPartitionInfo Interfaces JBAS-7992
- cglib

8.3. Entfernte Elemente

In dieser Release wurden keine Elemente entfernt.

9. In dieser Release behobene Probleme

Nachfolgend sehen Sie eine Liste der in dieser Release behobenen Probleme:

9.1. Allgemeine, behobene Probleme

JBPAPP-6987

Microcontainer Beans wurden beim Deployment als MBeans registriert. Diese MBeans sind optional und liefern Informationen zur Fehlerbehebung. Dieses Verhalten wurde jetzt für die Standardeinstellung abgestellt. Microcontainer Beans werden beim Deployment nicht mehr standardmäßig als MBeans registriert.

JBPAPP-4730

Die spezifische Kombination von JBoss Native, 64-bit Hardware und der **`-javaagent`** Java-Option führte bei Systemen dazu, dass sie bei Ausführung des **`run.sh`**-Skripts festhingen. Das Skript gab vom Nutzer gelieferte Parameter an Java, während ein Aufruf zur Prüfung der Systemarchitektur erfolgte. Das **`run.sh`**-Skript ist jetzt dahingehend bearbeitet, dass es vom Nutzer gelieferte Parameter bei diesem Aufruf ignoriert, so dass das Skript unter diesen Bedingungen nicht mehr festhängt.

JBPAPP-4683

Verwendete ein Management-Tool das ProfileService Management-Interface (wie es die eingebettete Konsole tut) um den ServiceBindingManager-Dienst zu aktualisieren, so war das Mapping der erhaltenen Information inkorrekt. Der erhaltene Wert für die "fixedHostName"-Property des Bindings wurde als "fixed Port" und umgekehrt angewendet. In den meisten Fällen sind beide Properties "false", so dass dies harmlos war. Wenn jedoch die Properties "true" waren, so führte dies zu einem inkorrekten Binding eim Server-Neustart, wenn die aktualisierten Werte angewendet wurden. Dies wurde korrigiert, und die Mappings werden jetzt wie erwartet angewendet.

JBPAPP-4471

Wenn Benutzertyp-gemappte Typen in ejbql Aggregatfunktionen wie MIN und MAX verwendet wurden, so wurde eine CCE-Ausnahme gemeldet. Dies wurde jetzt behoben und CMP-Felder von Typen mit Benutzertyp-Mapping können in Aggregatfunktionen in der SELECT-Klausel einer EJB-QL Anfrage erscheinen.

JBPAPP-3942

Das **`run.sh`**-Skript erstellt jetzt eine Datei zur Verfolgung von Prozessen und macht es einfacher, Prozess-IDs **`jboss.pidfile`** zu erhalten. Für weitere Informationen siehe JIRA.

JBPAPP-3674

Ein Status-Servlet-Fehler existierte für <distributable> Web-Apps. Der Cache Session-Manager hatte andere Attribute als der Web-Manager, wurde aber aufgrund fehlender Empfindlichkeit als dasselbe JMX-Objekt wahrgenommen. StandardManager verwendete 'activeSessions', während die geclusterte Version 'ActiveSessions' war. Der vom Servlet zum Rendern der Statusinformation verwendete Code kann jetzt Groß- und Kleinbuchstaben in den JMX-Attributnamen für einen Session-Manager aufspüren und handhaben.

JBPAPP-3171

JBPAPP-544 hat ein Sicherheitsproblem behoben, indem das Mapping für /status entfernt wurde, wodurch nur der sichere /web-console/status verfügbar blieb. Der ungesicherte /status wurde jedoch in JBPAPP-1146 wieder aufgenommen. Dies wurde behoben, indem das **`ROOT.war`** in der Security der Web-App konfiguriert wurde. Es löst das ursprüngliche Problem der Sicherheit und lässt /status intakt.

JBPAPP-2571

Wird der Microsoft SQL Server mit Microsoft JDBC Treibern 2.0 ausgeführt, verursacht dies instabile Builds in JBoss Messaging. Dieses Problem wird verursacht durch das **`Adaptive Buffering`**, was das Standardverhalten des Treibers ist, denn dies erlaubt nur das einmalige Lesen von großen Werten mittels der **`get<Type>Stream`**-Methode. Um dieses Problem im Moment zum umgehen, ändern Sie den ***`responseBuffering`***-Parameter der JDBC-Verbindungs-URL von **`adaptive`** auf **`full`** (wie es für Version 1.2 des JDBC-Treibers der Fall war).

```
<url>jdbc:sqlserver://[host];database=[database];responseBuffering=full;</ur
l>
```

JBPAPP-1774

Red Hat Enterprise Linux 5 Nutzer, die die JBoss Enterprise Application Platform rpm mittels **yum** installieren, müssen den Red Hat Enterprise Linux Extras Channel abonnieren, um die Java-Anforderungen der JBoss Enterprise Application Platform mit entweder dem Sun oder dem IBM Java Development Kit zu erfüllen.

9.2. Behobene Probleme mit Hibernate

JBPAPP-4579

Die Verwendung einer **`session.createSQLQuery`** führte zu einem Speicherleck, da die **`equals`**- und **`hashCode`**-Methoden in **`NativeSQLQuerySpecification`** nicht ordnungsgemäß implementiert waren und die gleichzeitige Verwendung von **`LRUMap`** dazu führte, dass die Map die Maximalgröße überschritt. Diese Probleme wurden behoben, und es kommt nicht mehr zu einem Speicherleck.

JBPAPP-4326

Die Änderungen für JBPAPP-4471 führten zu einem Problem wenn Nutzer nicht-JDK serialisierbare Klassen mit **`SerializableType`** mappten und versuchten, diese Werte in einem Cache der zweiten Ebene zu cachen. **`SerializableType`** meldete den Klassenlader von **`java.io.Serializable`** als den für die Deserialisierung zu verwendenden Klassenlader. Jedoch konnte dieser Klassenlader nicht in den Klassenlader oder nicht-JDK Klassen sehen.

Der Klassenlader aus **`org.hibernate.util.SerializationHelper.deserialize(InputStream, ClassLoader)`** versucht jetzt, den zu deserialisierenden Klassenlader aufzulösen. Ist dies nicht erfolgreich möglich, verwendet es den Kontextklassenlader aus dem aktuellen Thread und den Klassenlader, der die Hibernate-Klasse lädt.

JBPAPP-4224

Das Query Cache erforderte, dass die Session oder der Entity Manager geschlossen und erneut geöffnet wurden, ehe es wirksam wurde, da Hibernate den Zeitstempel einer gecachten Anfrage mit dem Zeitstempel der aktuellen Session vergleicht, wenn es eine Anfrage vom Cache abruft. Dies wurde jetzt behoben und Hibernate verwendet **`nextTimestamp()`** zum Abruf des aktuellen Zeitstempels statt den gecachten Zeitstempel.

JBPAPP-4183

`@CollectionId` beschreibt eine Bezeichnerspalte für eine Bag (z.B. eine idbag). **`nullable`** sollte explizit auf "false" gesetzt werden. Dies wurde behoben und **`nullable`** ist jetzt standardmäßig auf **`false`** eingestellt.

JBPAPP-4123

PostgreSQL ließ **`SchemaExport`** nicht fallen, wenn Einschränkungsnamen sich änderten. Hibernate gibt jetzt eine Kaskade heraus, wenn es versucht eine Tabelle fallen zu lassen, wodurch das Problem gelöst wird.

JBPAPP-4105

Hibernate verwendet **`StandardDialectResolver`**, um den Hibernate Dialekt zu erraten, wenn die **`hibernate.dialect`**-Property nicht in der Hibernate Properties-Datei aufgefunden wird, **`hibernate.cfg.xml`** oder **`persistence.xml`**. **`StandardDialectResolver`** gab **`SybaseDialect`** wieder, wenn der Datenbankname, den es aus den Datenbank-Metadaten extrahierte **`Sybase SQL Server`** oder **`Adaptive Server Enterprise`** ist. Da aber **`SybaseDialect`** veraltet ist, führte dies zu Problemen. Der **`StandardDialectResolver`** errät jetzt den korrekten zu verwendenden Dialekt, wenn **`hibernate.dialect`** nicht konfiguriert ist.

JBPAPP-4095

Beim Scrollen einer HQL-Anfrage mit Join Fetch an einer Collection bewegte sich **`FetchingScrollableResultsImpl.last()`** nicht zum letzten Ergebnis, wenn der Cursor sich hinter dem letzten Ergebnis befand. Stattdessen blieb der Cursor in der Position hinter dem letzten Ergebnis. Dies konnte zu **`org.hibernate.exception.GenericJDBCException`**: **`could not perform sequential read of results`** führen. Die Lösung für dieses Problem stellt nun sicher, dass der Cursor sich wie erwartet zum letzten Ergebnis bewegt.

JBPAPP-4088

War eine Spalte in Backticks (`) definiert, so schlugen **`@JoinTable`** und **`@JoinColumn`** Mappings mit **`AnnotationException`** fehl, wenn der Spaltenname mit Backticks referenziert wurde. Hibernate vergleicht jetzt den **`referencedColumnName`** mit dem in Anführungszeichen gesetzen Namen statt mit dem Tabellenspaltennamen.

JBPAPP-4022

`getColumnSpan` gab **`0`** wieder, wenn Entities mit **`OneToOne`**-Mappings verglichen wurden, selbst wenn dieselbe Entity verglichen wurde, was zu **`TypeMismatchException`**s führte. Der Algorithmus von **`getColumnSpan`** wurde geändert, so dass die Spaltenanzahl anhand des Bezeichners oder des eindeutigen Schlüsseltyps identifiziert werden kann.

JBPAPP-3946

`LikeExpression` handhabte das **`ignoreCase`**-Flag nicht ordnungsgemäß, wenn **`ignoreCase`** auf "false" gesetzt war. Die korrekte SQL **`property like ?`** wurde erstellt, aber das Flag in **`getTypedValues`** wurde nicht verwendet, und es wurde stets ein Wert in Kleinbuchstaben produziert. Dieses Problem wurde behoben und Kleinbuchstaben oder der Originalfall werden verwendet, wenn **`ignoreCase`** auf "false" gesetzt ist.

JBPAPP-3737

In einer stateless Session laden Anfragen Objekte in einem zweiphasigen Vorgang: In der ersten Phase wird ein temporärer Persistenzkontext mit leeren Objekten bevölkert. In der zweiten Phase werden die Mitgliedsdaten der Objekte aus der Datenbank gelesen. Enthält ein Objekt eine Assoziation oder eine Collection, so führt die Anfrage einen rekursiven Aufruf an die get()-Methode der Session durch. Dies bereinigt den temporären Persistenzkontext.

Enthielt das übergeordnete Objekt andere Assoziationen, die als Teil der zweiten Phase gelesen werden sollten, so meldete Hibernate eine Ausnahme, da die Assoziationen nicht im Persistenzkontext gefunden werden konnten.

Dies wurde mittels einer neuen Methode behoben:

org.hibernate.engine.PersistenceContext.isLoadFinished(). Diese Methode teilt StatelessSession mit, wann ein temporärer Persistenzkontext bereinigt werden soll.

JBPAPP-3565

Ein Fehler im **org.hibernate.cfg.OneToOneSecondPass**-Code führte manchmal zu einer **org.hibernate.PropertyValueException** wenn ein One-to-One Mapping verarbeitet wurde. Die Ausnahme erfolgte deshalb, weil wenn JOINs **otherSideProperty**, **otherSideJoin** alte Daten enthielten statt dass sie Null waren. Der **org.hibernate.cfg.OneToOneSecondPass**-Code wurde korrigiert.

JBPAPP-3487

Hibernate generierte in einigen Situationen den falschen Alias in der Table-pre-class Vererbungsstrategie. (Beispiele finden Sie in der JIRA.) **AbstractCollectionPersister** verwendet jetzt Schlüsselspaltenaliasse, die von **Column.getAlias(Dialect,Table)** generiert werden, so dass das Table-pre-class Vererbungsstrategie-Mapping korrekt ist.

JBPAPP-2440

Konnte ein Cache-Provider nicht gefunden werden, so wurde ein NoClassDefFoundError gemeldet, der folgende Nachricht enthielt:

```
net/sf/ehcache/CacheException
```

Konnte ein Connection-Provider nicht gefunden werden, so wurde eine **HibernateException** mit folgender Nachricht gemeldet:

```
Could not instantiate connection provider: " + providerClass
```

Es erfolgen jetzt informativere Nachrichten, wenn diese Fehler auftreten.

JBPAPP-2276

Die Iterationsreihenfolge von **HashMaps** und **HashSets** für JDK 6 verursacht unterschiedliche Spaltenreihenfolgen in Union-Klauseln oder Union-Unterklassen abhängig davon, ob JDK 5 oder 6 verwendet wird. Da die Abweichung der Spaltenreihenfolgen über die Union-Klauseln hinweg konsistent ist, sind die daraus folgenden Abfragen gültig, allerdings kann diese Veränderung möglicherweise die Leistung beeinträchtigen. Die Implementierung von HashSet wurde für die identische Performance zwischen den verschiedenen JDKs zu LinkedHashSet geändert.

JBPAPP-2275

Hibernate konnte in der Vergangenheit unter JDK 6 nicht kompiliert werden, weil die folgenden Klassen JDK 6 Interfaces nicht vollständig implementierten:

- **org.hibernate.jdbc.ResultSetWrapper**
- **org.hibernate.lob.BlobImpl** to implement **java.sql.Blob**
- **org.hibernate.lob.ClobImpl**
- **org.hibernate.lob.SerializableBlob**
- **org.hibernate.lob.SerializableClob**

Es kam zu einem NoSuchMethodError wenn eine Applikation eine Methode erforderte, die in diesen Klassen fehlte. Die fehlenden Methoden wurden nun implementiert, so dass dieser Fehler nicht mehr auftritt.

9.3. Behobene Remoting Probleme

JBPAPP-3709

Wenn **`org.jboss.remoting.marshal.MarshallerLoaderHandler`** eine Anfrage für eine Klasse erhielt gab es eine Instanz von **`org.jboss.remoting.loading.ClassBytes`** wieder. Konnte **`MarshallerLoaderHandler`** die gewünschte Klasse nicht finden, dann hatte das wiedergegebene **`ClassBytes`**-Objekt einen Nullwert für den Klassen Byte Array. Allerdings prüft der **`org.jboss.remoting.loading.ClassByteClassLoader`** nicht auf die Möglichkeit eines Null Byte Array, so dass es zu einer **`NullPointerException`** kam. JBREM-1184, welches zu diesem Problem führte, wurde behoben.

9.4. Behobene Probleme mit Seam

JBPAPP-4876

Bei der Einspeisung einer jBPM **`ProcessInstance`** in eine Seam Komponente wurde die wiedergegebene Prozessinstanz für "auto-save" registriert. Dies führte zu einem Problem mit der Performance, da der jBPM ManagedContext alle Instanzen in der **`autoSaveProcessesList`** speichert, wenn er schließt. Der Code für die Wiedergabe einer **`ProcessInstance`** bei Einspeisung wurde geändert, und die Prozessinstanz ist nicht mehr für "auto-save" registriert.

JBPAPP-4685

Eine Eingabe-Bereinigungsschwachstelle wurde bei der Art und Weise, auf die JBoss Seam bestimmte parametrisierte JBoss Expression Language (EL) Ausdrücke verarbeitete gefunden. Ein Remote-Angreifer konnte diese Schwachstelle zur Ausführung arbiträren Codes via URL nutzen, die angehängte, speziell gefertigte Expression Language Parameters enthielten, die bestimmten Applikationen basierend auf das JBoss Seam Framework geliefert wurden. Hinweis: Ein ordnungsgemäß konfigurierter und aktivierter Java Security Manager würde die Ausnutzung dieser Sicherheitslücke verhindern. (CVE-2010-1871)

Red Hat möchte sich bei Meder Kydyraliev vom Google Security Team für die Meldung dieses Problems bedanken.

JBPAPP-4577

Eine Änderung an Seams **`getInstanceFromFactory`**-Methode zur Behebung eines Nebenläufigkeitsproblems führte bei einer ganz spezifischen Kombination von Umständen bei einigen Applikationen zu einer Deadlock-Situation. Die ursprüngliche Problembehebung von **`getInstanceFromFactory`** wurde überarbeitet, und der Fehler wurde seitdem nicht mehr beobachtet.

JBPAPP-4568

Seam wurde mit einer Kopie von **`activation.jar`** in seinem **`/lib`**-Verzeichnis distribuiert. Diese Datei wird auch vom JDK 6 bereitgestellt. Die Kopie in Seam wurde entfernt.

JBPAPP-4509

Rich:comboBox Komponenten in Seam Apps, die mit **`seam-gen`** erstellt wurden fehlte die Drop-Down-Pfeilgrafik. Dies wurde behoben, indem das kaskadierende Style Sheet bearbeitet wurde, das die Anzeige der Komponente in einem Webbrowser steuert. Die Pfeilgrafik wird jetzt korrekt angezeigt.

JBPAPP-4317

Seam würde fehlschlagen wenn zwei Komponenten mit demselben Klassennamen im Klassenpfad gefunden würden; folgender Fehler würde gemeldet : "**`java.lang.IllegalStateException: Two components with the same name and precedence -`**", gefolgt von der Komponente und Klassennamen. Seam wurde aktualisiert und ignoriert jetzt doppelte Klassen. Trifft es auf doppelte Komponenten mit demselben Klassennamen, so wird die erste auf die es trifft geladen und jede weitere Komponente im Klassenpfad mit demselben Klassennamen wird ignoriert.

JBPAPP-4273

`MockServletContext` war nicht darüber informiert, dass die JBoss Enterprise Application Platform 5 das Virtual File System verwendete. Es versuchte daher auf URIs als Dateien zuzugreifen. **`MockServletContext`** erkennt jetzt den Web Applikation Root und ruft den Pfad der URL-Klasse ab, wodurch sichergestellt wird, dass das Protokoll keinen Teil des **`java.io.File`** Constructor-Parameters bildet.

JBPAPP-4259

Seam-gen Vorlagen verwendeten **`resource://`** für Objekte, die nicht im Klassenpfad waren. Dies führte beim Versuch eine RichFaces Vorlage (XCSS) in eine CSS-Datei umzuwandeln zu einem Fehler. Seam-gen verwendet **`resource://`** nicht mehr, wodurch das Problem behoben wurde.

JBPAPP-4015

Beim Hot Deployment von mehrere WAR-Dateien enthaltenden EARs funktionierte die erste Webanwendung auf die zugegriffen wurde ordnungsgemäß, Anfragen an nachfolgende Webanwendungen jedoch führten zu einer **`NullPointerException`**. Dies war der Fall, weil der Applikationskontext nicht eingestellt war, wenn der Hot Deployment Filter lief. Jeder Vorgang, der mit einem Webanwendungskontext zu tun hat, prüft jetzt auf den aktuellen **`javax.servlet.ServletContext`**, wodurch das Problem gelöst ist.

JBPAPP-4013

Bei der Verwendung der Web Services Seite des Seambay-Beispiels im Internet Explorer 8, pflanzen sich die in den Benutzernamen- und Passwortfeldern eingegebenen Werte wegen eines undefinierten JavaScript-Objekts nicht zum Anfragebereich fort. Dies wurde korrigiert.

JBPAPP-4012

Das Tasks-Beispiel funktionierte nicht ordnungsgemäß im Internet Explorer 8. Wird "Undo this Task" auf der "Resolved Tasks"-Seite gewählt, so kehrt die Task zur Tasks-Seite zurück, verschwindet jedoch nicht von der "Resolved Tasks"-Seite. Dieses Problem wurde durch Abschalten des Ajax Anfragen-Cachings behoben.

JBPAPP-4010

Ressourcen vom Ressource-Servlet, wie etwa CAPTCHA Abbildungen, wurden vom Browser gecacht und wurden wegen inkorrekter Response-Header in **`org.jboss.seam.captcha.CaptchaImage`** nicht korrekt neu gerendert. Die Response-Header wurden korrigiert, so dass dies nicht mehr vorkommt.

JBPAPP-4009

In Seam Mail waren Anhänge nicht immer inline enthalten, so dass Anhänge in manchen Mail Clients nicht sichtbar waren. Der Patch für dieses Problem wurde von JBSEAM-4630 rückportiert und Anhänge sollten nun sichtbar sein.

JBPAPP-4008

Wurde **`ELSupport.coerceToType`** mit einer **`BigDecimal`**-Instanz als dem ersten Parameter 'obj' und **`BigDecimal.class`** als Typ aufgerufen, so wurde **`BigDecimal`** in ein "double" und wieder in ein **`BigDecimal`** zurück konvertiert, was zu einem anderen Wiedergabewert führte.

Darauf wurde Upstream eingegangen, und es wurde in JBoss Enterprise Application Platform 5.1.0 durch Änderung der Seam Abhängigkeit von JBoss-EL to 1.0_02.CR5 behoben.

ELSupport.coerceToType gibt jetzt einen Eingabewert wieder, wenn er zum gegebenen Typ zuweisbar ist.

JBPAPP-4007

`org.jboss.seam.transaction.EjbSynchronizations` stimmte nicht mit Java Spezifikationen überein. Wie in http://java.sun.com/blueprints/guidelines/designing_enterprise_applications/transaction_management/qanda/index.html dargestellt, können SessionSynchronization implementierende Klassen **`TransactionsAttribute TransactionAttributeType.SUPPORTS`** nicht haben. **`TransactionsAttributeType.SUPPORTS`** wurde durch **`TransactionsAttributeType.REQUIRED`** ersetzt.

JBPAPP-4006

`org.jboss.seam.core.Interpolator` meldete keine Ausnahmen und druckte Warnungen mit beschränkten Informationen. Ausnahmen werden jetzt gemeldet und Warnungen sind informativer.

JBPAPP-4005

Seam JMS-Support funktionierte nicht nach einem Failover zu einem Slave-Node, da das **`topicConnection`**-Feld im **`SubscriptionRegistry`**-Objekt ungültig wurde, wenn die Verbindung zum Master-Node fehlschlug. Nachfolgende Aufrufe von **`SubscriptionRegistry.subscribe()`** schlugen mit **`SpyJMSException`** fehl, und JMS-Abonnements funktionierten nicht mehr, selbst wenn der Master-Node wieder funktionstüchtig war. Dieses Problem ist jetzt behoben.

JBPAPP-4004

Es gab einen typografischen Fehler in der vom **`org.jboss.seam.async.AsynchronousExceptionHandler`** protokollierten Nachricht,

wenn "Exeception thrown while executing asynchronous call" gemeldet wurde. "Exeception" sollte eigentlich "Exception" sein. Dies ist jetzt behoben.

JBPAPP-4003

Die Anforderungen von JBoss Enterprise Application Platform 4.x und JBoss Enterprise Application Platform 5.x sind verschieden und **`messages.properties`** auf Seitenebene im **`ResourceLoader`** müssen in **`META-INF/classes`** gespeichert werden oder sie funktionieren nicht. Die Dokumentation behandelt dies nicht explizit. Das Problem wurde behoben.

JBPAPP-4002

`Initialization.redeploy` enthielt Code, der eine neue Sperre erstellt und eine sofortige Sperrung durchführte. Dies bot keinerlei Schutzebene und konnte im Debug-Modus zu Problemen führen. Dies wurde behoben, indem die Sperre komplett entfernt wurde.

JBPAPP-3855

Der SeamSpace **`testCreateBlog`** schlug mit **`NullPointerException`** auf IBM JDK 1.6 fehl. Die getIdentifier-Methode verhält sich jetzt korrekt auf JDK 1.6.

JBPAPP-3769

In Profilen außer **`production`** setzte **`MockResponseStateManager`** **`ResponseStateManager#isPostback`** nicht außer Kraft. Dies bedeutete, dass sowohl Faces als auch nicht-Faces Anfragen als Postbacks evaluiert wurden. Eine **`isPostback(FacesContext context)`**-Methode wurde in **`org.jboss.seam.mock.MockResponseStateManager`** hinzugefügt, die die alte Methode außer Kraft setzt und NonFacesRequest die Erkennung von Postback gestattet.

JBPAPP-3713

JAX-RS Webdienste, die innerhalb einer EJB-Transaktion ausgeführt werden, können Seam HTTP-Authentifikation nicht nutzen. HTTP-Anfragen an einen Webdienst, der durch **`AuthenticationFilter`** geschützt ist, spalten sich in zwei **`ContextualHttpRequests`** auf. In der zweiten ist ein Lookup von EjbSynchronization Beans nicht möglich, so dass **`Component.getInstance("org.jboss.seam.transaction.synchronizations", ScopeType.EVENT)`** mit **`NoSuchEjbException`** fehlschlägt.

Dieses Problem wurde behoben.

JBPAPP-3525

`s.tld` in **`jboss-seam.ui.jar/MANIFEST.MF`** enthielt Attributeinträge, die gemäß dem XML-Schema für den JSP Taglibrary Deskriptor inkorrekt definiert waren. Dies wurde im Upgrade zu Richfaces 3.3.1.SP1 behoben.

JBPAPP-2384

Das Seam Chatroom Beispiel meldete eine **`EJBTransactionRolledbackException`** wenn ein Nutzer versuchte sich anzumelden. Diese Ausnahme hatte keinen Einfluss auf die Funktionalität der Anwendung und wurde mittels eines Updates von JBoss Cache behoben.

9.5. Behobene Security Probleme

JBPAPP-4144

Auf das /status servlet konnte von **`http://localhost:8080/status?full=true`** ohne Authentifikation zugegriffen werden. (CVE-2008-3273)

Dieses Problem ist behoben, und das Servlet erfordert jetzt Authentifikation.

9.6. Behobene Applikationsserver Probleme

JBPAPP-4244

Der Parsing und Deploying Mechanismus für **`hibernate.cfg.xml`** war instabil. Dies führte dazu, dass reguläre Hibernate Property Namen nicht benutzt werden konnten, weil Properties zur **`org.jboss.hibernate.jmx.Hibernate class`** gemappt werden musstem. Der ursprüngliche Satz geparster Schlüssel/Werte wird jetzt mittels einer neuen **`setProperties(Set<BaseNamedElement>)`** -Methode in die **`o.j.h.hmx.Hibernate`**-Objekte eingefügt. Die **`start()`**-Method baut dann aus dem Satz von Properties die Konfiguration.

9.7. Behobene Dokumentationsprobleme

JBPAPP-5036

Mehrere typografische Fehler im Seam Referenzhandbuch wurden korrigiert.

JBPAPP-4849

Der Abschnitt *Running as a Service on Windows* im *Installationshandbuch* listete inkorrekte lange und kurze Dienstnamen und einen inkorrekten Speicherort für das **`service.bat`**-Skript. Dies wurde behoben.

JBPAPP-4391

Die RuleBasedPermissionResolver-Anforderungen für Drools 5 waren veraltet. Dies führte dazu, dass inkorrekte Informationen in der Dokumentation präsentiert wurden. Die veralteten Informationen wurden durch aktuelle, akkurate Inhalte ersetzt.

JBPAPP-4390

Das Seam Referenzhandbuch enthielt doppelte Informationen, die jetzt entfernt sind.

JBPAPP-4389

Im 'Charting'-Abschnitt im Seam Referenzhandbuch fehlten Informationen zu <p:chart>-Tags. Dies wurde mittels einer Beschreibung korrigiert; 'displays a chart already created in Java by a Seam component.'

JBPAPP-4387

Quellcode-Blocks der Seam-Referenzdokumentation besaßen keine Syntaxmarkierung, so dass der Code schwer lesbar war. Dieser Code wurde vollständig mit der korrekten Markierung aktualisiert.

JBPAPP-4386

Im Seam Referenzhandbuch fehlten Callout-Nummern in Beispiel-Codeblocks. Dadurch war es schwierig, eine Verbindung zwischen dem Code und den begleitenden Erklärungshinweisen herzustellen. Dies wurde durch die Einführung von Callouts behoben.

JBPAPP-3818

"Using Seam with JBoss Tools" im *Seam Referenzhandbuch* wurde nicht für die Verwendung mit der JBoss Enterprise Application Platform überarbeitet und sollte als überholt angesehen werden.

Die Fehler in der Dokumentation wurden behoben, und es fand eine Aktualisierung statt.

9.8. Behobene Probleme bei Messaging

JBPAPP-4135

`ServerConsumerEndpoint.localClose()` meldete beim Shutdown eine **`NullPointerException`**. Dies wurde mittels einer aktualisierten Version von JBoss Messaging behoben.

JBPAPP-3157

Werden Unicode Multibyte Zeichen in einem Message Selector verwendet, so meldet JBoss Messaging einen TokenMgrError.

Dieses Problem wurde nun behoben und Multibyte-Zeichen sind jetzt gestattet.

JBPAPP-3003

Wenn JBoss Transactions von Version 4.4.0 auf Version 4.6.1 aktualisiert wird, schlägt der JBoss Messaging **`XAResourceRecoveryTest`** fehl. Dieser Test soll einen Fehler während des Festschreibungsvorgangs simulieren, um den Recovery Manager dazu zu veranlassen, die Transaktion wiederherzustellen, doch mit JBoss Transactions 4.6.1 wird die Transaktion nicht wiederhergestellt.

Dieses Problem ist jetzt behoben, und ein neuer Test wurde erstellt, der die Änderungen im Transaktionsverhalten handhabt.

9.9. Behobene Probleme mit Web Services

JBPAPP-4501

Tests haben Probleme mit der Art und Weise aufgedeckt, auf die die Enterprise Application Platform 5.1.0 den Transfer zahlreicher großer Dateien ohne aktiviertes MTOM handhabt. Kunden sollten beachten, dass MTOM zur Übertragung großer Dateien aktiviert sein sollte und diese Art von Dienst eine große Menge an Speicher benötigt. Falls Sie sich jedoch gegen die Verwendung von MTOM entscheiden, so können Sie folgende Implementierung durchführen, damit Ihr Applikationsserver nicht abstürzt:

- Konfigurieren Sie den eingeschränkten Horcher für diesen Dienst
- Weisen Sie mehr Speicher zu und konfigurieren Sie die richtige Garbage Collection (GC) Policy für eine Java Virtual Machine

JBPAPP-4243

Das Webdienst-Kontextbeispiel meldete eine **`ClassCastException`**. Das Beispiel wurde gepatcht und funktioniert jetzt ordnungsgemäß.

JBPAPP-3785

Wurde eine Methode mit einem zusätzlichen Namespace für einen Header-Parameter nach einer Methode ohne diesen zusätzlichen Namespace aufgerufen, so wurde der falsche Namespace in der SOAP-Operation verwendet, wenn die Methode serialisiert wurde. Dieses Problem wurde behoben, indem die Art und Weise auf die Transitionen zwischen Java Objekten und XML erfolgen, korrigiert wurde, so dass die endgültige SOAP-Nachricht mit korrekten Namespaces generiert wird.

JBPAPP-3019

Der doc/examples/jboss-web-services-examples Kontext führte zu einer Reihe von Ausnahmen. Dieser Kontextfehler bedeutete, dass die JBoss Web Services Beispiele nicht ordnungsgemäß funktionierten. Der Beispielcode wurde gemäß JBPAPP-2995 geändert, und die Beispiele funktionieren jetzt korrekt.

JBPAPP-2162

Sun JAXB akzeptiert stillschweigend Meldungen mit nicht-fatalen Fehlern, obwohl diese zurückgewiesen werden sollten. Die Fehlerbehebung für JBPAPP-2114 korrigiert dies, so dass die fehlerhaften Nachrichten zurückgewiesen werden.

9.10. Behobene Probleme mit RESTEasy

JBPAPP-4061

Eine **`IndexOutOfBoundsException`** wurde in der **`POJOResourceFactory`** gemeldet, weil das Interface anstatt der Implementierung annotiert war. Support für annotierte Interfaces wurde hinzugefügt.

9.11. Behobene Clustering Probleme

JBPAPP-4887

Wird eine JPA oder Hibernate verwendende Applikation dahingehend konfiguriert, dass Sie ein geclustertes Cache der zweiten Ebene verwendet, so wird eine explizite Abhängigkeit vom CacheManager Service des Applikationsservers erstellt. Diese Abhängigkeit wird nicht im Abhängigkeitssystem des Applikationsservers gespeichert, so dass beim Stoppen der Plattform manchmal der CacheManager vor der Applikation herunterfährt. Dies führt zu einer WARN-Meldung im Protokoll, wenn die Applikation herunterfährt. Dieses Problem führt nur zu Warnungen im Fehlerprotokoll und wurde mittels eines HA Server Cache Upgrade behoben.

9.12. Behobene Cache Probleme

JBPAPP-4254

ha-server-cache-jbc has been upgraded to 2.0.3.Final.

9.13. Behobene Connector Probleme

JBPAPP-4307

Bei der Verwendung von **`SystemMemoryUsageLoadMetric`**, priorisiert mod_cluster Versandanfragen ("dispatching requests") an eine Maschine, die einen höheren Systemspeicher als andere Maschinen verwendet. Die Metrik gab dessen 'Last' als 'freien Speicher' anstatt des erwarteten 'verwendeten Speicher' wieder. Dies wurde korrigiert und der Fehler tritt nicht mehr auf.

MODCLUSTER-113

`org.jboss.modcluster.demo.servlet.ThreadCountLoadServlet` wurde aus mod_cluster entfernt, ist jedoch nach wie vor in der zu **`load-demo.war`** gehörenden **`web.xml`**-Datei festgelegt. Dies führt zu Deployment-Fehlern. Das Problem wurde behoben und das nicht existente Servlet wurde entfernt.

10. Bei dieser Release bekannte Probleme

Nachfolgend sehen Sie eine Liste der zum Zeitpunkt der Release bekannten Probleme.

10.1. Nicht unterstützte JARs

Die folgenden Bibliotheken sind in dieser Version der JBoss Enterprise Application Platform enthalten, werden jedoch nicht unterstützt.

- **`client/hornetq-core-client.jar`**
- **`client/hornetq-jms-client.jar`**

Der **`hornetq`**-RPM wird in dieser Release ebenfalls nicht unterstützt.

10.2. Allgemeine, bekannte Probleme

JBPAPP-5110

Der grafische Installer gestattet es dem Nutzer eine benutzerdefinierte JAAS Security Domain zur Sicherung von Domains und Invokers festzulegen, aber sichert die Tomcat Konsole immer mit der JAAS Security Domain "jmx-console", selbst wenn sie nicht existiert. Um eine benutzerdefinierte JAAS Security Domain zu verwenden, legen Sie den Namen während der Installation im "JMX Security"-Bildschirm fest. Ist der Installer fertig, bearbeiten Sie die **`jboss-as/server/$PROFILE/deploy/ROOT.war/WEB-INF/jboss-web.xml`**-Datei. Im **`<security-domain>`**-Element ersetzen Sie den Namen "jmx-console" durch den Namen der Security Domain, die Sie während der Installation verwendet haben. Dies sichert die Tomcat Konsole mit derselben benuterdefinierten JAAS Security Domain, die die Admin Konsole sowie andere Konsolen und Invoker sichert, die Sie während der Installation gewählt haben.

JBPAPP-5014

Das Web-Profil beinhaltet inkorrekter Weise eine Recovery-Einstellung für JBoss Messaging, die nicht Teil des Web-Profils ist. Dies führt zu **`ClassNotFoundException`**s in **`server.log`**.

Um dieses Problem zu umgehen, entfernen Sie die folgenden Zeilen aus **`jbossts-properties.xml`**:

```
<property
name="com.arjuna.ats.jta.recovery.XAResourceRecovery.JBMESSAGING1"

value="org.jboss.jms.server.recovery.MessagingXAResourceRecovery;java:/Defa
ultJMSProvider"/>
```

JBPAPP-4542

Die Speicherorte von **`jbosssx.jar`**, **`jboss-javaee.jar`**, **`jboss-security-spi.jar`** und **`jbosssx-server.jar`** haben sich in dieser Release aufgrund eines Security Fix geändert. In dieser Release befinden sie sich in **`jboss-as/lib/`**. Skripte, die vom Speicherort von **`jbosssx.jar`** abhängen, sollten entsprechend dieser Änderung aktualisiert werden.

JBPAPP-4665

Das Deployment von Resteasy-guice Applikationen schlägt wegen einer java.lang.SecurityException fehl. Eine der folgenden ähnliche Fehlermeldung wird angezeigt: java.lang.SecurityException: class "org.jboss.resteasy.examples.guice.hello.DefaultGreeter$$FastClassByGuice$$70fd68d0"'s signer information does not match signer information of other classes in the same package"

Ein Patch für dieses Problem wurde zusammen mit JBoss Enterprise Application Platform 5.1.0 veröffentlicht und kann von der Red Hat Support-Seite heruntergeladen werden.

JBPAPP-4456

Andere Agent-aktivierte Tools könnten auf eine Klasse stoßen, die ein Feld, eine Methode, einen Konstruktor oder anderes noch nicht von Javassist geladenes enthält. In diesem Fall wird eine **`javassist.NotFoundException`** gemeldet, wenn der Advisor diese Methode zu manipulieren versucht. Diese Ausnahme sollte ignoriert und das Feld als nicht existent betrachtet werden.

JBPAPP-4130

Ein **`org.jboss.ejb3.stateless.StatelessDelegateWrapper`** kann keinem **`org.jboss.system.ServiceMBean`** zugewiesen werden, wenn ein Monitor mit **`org.jboss.console.plugins.monitor.CreateThresholdMonitorServlet`** erstellt wird. Der Monitor wird als Attribut statt als ein <depends> erstellt und schlägt bei einem Neustart daher fehl. Sie umgehen dieses Problem, indem Sie die erstellte **`*-service.xml`** manuell ändern, um dem Monitor das ordnungsgemäße Deployment nach dem Neustart zu ermöglichen.

JBPAPP-4041

Der **`ServiceMetaDataParser.parseValueFactoryParameter()`** berücksichtigt nur das erste Child des **`<parameter>`**-Elements. Ist das **`<null/>`** Child-Element also von Zeilenumbrüchen ('Enter' oder Leerzeichen) umgeben, so wird der Node als Textwert behandelt und der Parameter wird später nicht ordnungsgemäß substituiert. Folgendes führt zu einem Fehler:

```
<parameter>
      <null/>
      </parameter>
```

Sie umgehen das Problem indem Sie sicherstellen, dass sich kein Leerzeichen in **<parameter>**-Elementen befindet:

```
<parameter><null/></parameter>
```

JBPAPP-3929

Wenn **java.sql.Date.valueOf** versucht Daten des Formats yyyy-mm-dd zu parsen, meldete der TCK-Test eine **java.lang.IllegalArgumentException**. Der Grund hierfür war eine Regression in der aktuellsten Sun JVM, Sun JDK 1.6.0_18-b07 (weitere Informationen finden Sie unter http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6898593). Die Lösung für dieses Problem besteht in einem Downgrade zu Sun JDK 1.6.0_17-b04.

JBPAPP-3766

Jedes Mal, wenn ein **org.jboss.mail.MailService** konfiguriert und an ein JNDI gebunden wird, werden die **SessionObjectFactories**-Properties überschrieben. Das bedeutet, dass wenn mehrere **MailService**-Konfigurationen existieren, alle an JNDI gebundenen **MailService**s die Properties des zuletzt konfigurierten **MailService** in ihrer Session haben.

JBPAPP-3308

In isolierten Deployments wird eine **ClassNotFoundException** gemeldet, wenn der Applikationsserver versucht, ein mit dem **Timer** assoziiertes **info**-Objekt zu deserialisieren . Dies geschieht, weil der falsche Klassenlader (**threadContextClassLoader**) zur Deserialisierung des Objekts verwendet wird.

JBPAPP-3280

Beim Zugriff auf die Admin-Konsole werden **WARN**-Nachrichten mit dem Text "**Cannot get the process table information without native support**" in den Protokollen generiert. Sie können diese Nachrichten problemlos ignorieren. In einer zukünftigen Release der Admin-Konsole werden diese Nachrichten in **INFO** geändert .

JBPAPP-3133

Ein Fehler bei JVM-Implementierungen bedeutet, dass die JVM abstürzen kann wenn während des Hot Deployments ein unvollständiges Deployment durch eine neue Version des Deployments überschrieben wird. Ein Deployment kann einen Formatierungsfehler in einem Deployment Deskriptor enthalten. Beim Deployment kommt dieser Fehler zum Tragen und das Deployment schlägt fehl. Wird ein korrigiertes Deployment über das fehlerhafte Deployment kopiert um es zu ersetzen, so ist es möglich, dass die JVM während des Hot Deployments der neuen Version abstürzt. Sie umgehen das Problem, indem Sie das alte, unvollständig deployte Archiv entfernen, ehe Sie das neue, korrigierte Deployment in das **deploy**-Verzeichnis kopieren.

JBPAPP-3036

Das **jboss_init_hpux**-Skript übernimmt keine Umgebungsvariablen, wenn es in der GNU Bash Shell ausgeführt wird. Dies hängt zusammen mit JBPAPP-2036: https://jira.jboss.org/jira/browse/JBPAPP-2306.

JBPAPP-3035

Die **-e**- und **-H**-Argumente des **shutdown.sh**-Skripts können nicht dazu verwendet werden, die JVM direkt zu beenden.

JBPAPP-2998

Wird der Server über ein Desktop-Symbol gestartet, so wird der standardmäßige Javasatz der Maschine verwendet. Dies kann zu Ausnahmen führen, wenn eine andere Java-Version als JDK 1.6 als Standard eingestellt ist. Benutzer sollten sicherstellen, dass JDK 1.6 als standardmäßige Java-Version eingestellt ist, ehe Sie versuchen die JBoss Enterprise Application Platform über das Desktop-Symbol zu starten.

JBPAPP-2894

Das Setzen der **hibernate.bytecode.provider**-Systemeigenschaft in **jpa-deployers-jboss-beans.xml** ist unzuverlässig. Fügen Sie zur Umgehung dieses Problems **-Dhibernate.bytecode.provider=cglib** zur **$JAVA_OPTS** in **jboss-as/bin/run.conf** hinzu.

JBPAPP-2871

Es wurden eine hohe CPU-Auslastung sowie eingeschränkte Leistung und Transaktionsdurchsatz beobachtet, wenn MySQL 5.0.41 mit optimierten Einstellungen wie in JBQA-2610 beschrieben verwendet wird. Wir empfehlen, auf MySQL 5.0.86 zu aktualisieren und optimierte Einstellungen wie unter JBQA-2610 beschrieben anzuwenden, um die CPU-Auslastung zu begrenzen und die Leistung zu erhöhen.

JBPAPP-2598

Sobald das JBAS-7049-Problem allerdings wie beschrieben umgangen wird, ergibt sich dadurch ein neues Problem. Der Start eines Servers, auf dem unter Verwendung von Open JDK 6 der Sicherheits-Manager ausgeführt wird, wird immer noch fehlschlagen, diesmal mit einem "access denied"-Fehler. Derzeit gibt es keine Möglichkeit, dieses Problem zu umgehen.

JBPAPP-2590

Beim Einsatz von IBM JDK 6 besteht ein Problem im **policy.provider**, der in **${JAVA_HOME}/jre/lib/security/java.security** definiert ist. Standardmäßig wird **org.apache.harmony.security.fortress.DefaultPolicy** verwendet, und dies sollte **policy.provider=sun.security.provider.PolicyFile** sein. Führen Sie zur Behebung dieses Problems diese Anpassung manuell aus.

JBPAPP-2576

Der MySQL JDBC-Treiber implementiert XA Recovery derzeit nicht einwandfrei.

JBPAPP-2027

Der Server-Manager funktioniert beim Einsatz von Open JDK 6 nicht ordnungsgemäß, da eine **NullPointerException**-Überprüfung in Open JDK 6 fehlt. Um dieses Problem zu umgehen, kommentieren Sie die java.security.debug-Anweisung in der **imports/server-config.xml**-Datei aus.

JBPAPP-2027

Die IBM-Distribution von JDK 6 unterstützt das **`SSLv2Hello`**-Protokoll nicht und generiert einen ERROR [AbstractKernelController], wenn dies verwendet wird. Es wird derzeit davon abgeraten, dieses Protokoll zu verwenden.

10.3. Bekannte Installer-Probleme

JBPAPP-5116

Wird eine benutzerdefinierte JAAS Security Domain während des grafischen Installationsprozesses erstellt, so fährt die JAAS Security Domain mit der Verwendung der **`jmx-console-users.properties`** und **`jmx-console-roles.properties`** fort, statt ihre eigenen benannten Nutzer- und Rollen-Properties-Dateien zu erstellen.

JBPAPP-4262

Nachdem alle Dateien während der Installation auf Solaris 10 (sowohl 32-bit als auch 64-bit Architekturen) extrahiert wurden, zeigt die Konsole eine unnötige Passwortanfrage an. Die Installation wird erst fortgesetzt, wenn der Nutzer die **`Enter`**-Taste drückt.

10.4. Bekannte Probleme mit Hibernate

JBPAPP-4926

Kaskadierende Tabellen, die autoinkrementierte Primärschlüssel enthalten, wurden nicht ordnungsgemäß gespeichert, da Tabellen nicht in der richtigen Reihenfolge in der Speicherwarteschlange verarbeitet wurden.

Eine Tabelle (A) hat einen zugewiesenen Primärschlüssel. Eine zweite Tabelle (B) ist mit Tabelle A verknüpft, die Kaskadierung ist aktiviert, und es wird ein autoinkrementierter Primärschlüssel verwendet. Beim Versuch Tabelle B zu speichern stellt Hibernate fest, dass Tabelle A zuerst gespeichert werden sollte und fügt diese der Speicherwarteschlange hinzu. Hibernate versucht dann sofort Tabelle B zu speichern. Dieser Speichervorgang schlägt fehl, da Tabelle B eine Primärschlüsselreferenz von Tabelle A benötigt, die jedoch Null ist, während sich diese in der Speicherwarteschlange befindet.

Um dieses Problem zu umgehen, deaktivieren Sie die Tabellenkaskadierung und stellen Sie sicher, dass die Tabelle mit dem zugewiesenen Primärschlüssel gespeichert wird, ehe Sie versuchen eine Tabelle mit autoinkrementiertem Primärschlüssel zu speichern.

JBPAPP-4895

Wenn eine **`refresh()`**-Methode sofort vor einem **`insert()`** aufgerufen wird und das Caching der zweiten Ebene aktiviert ist, so wird die Entity ins Cache der zweiten Ebene eingefügt. Wird **`refresh()`** jedoch nicht erfolgreich festgeschrieben, so werden die gecachten Daten nicht automatisch geräumt. Dies geschieht, weil **`refresh()`** den Entity-Status nicht verfolgt. Da der **`refresh()`**-Vorgang in der Regel für generierte Properties verwendet wird, besteht eine Problemumgehung in diesem Fall in der Verwendung der **`@Generated`**-Annotation. Sie können das Problem auch umgehen, indem Sie den Cache-Eintrag manuell räumen.

JBPAPP-4738

Der aktualisierte JDBC 3.0 Treiber für den MS SQL Server 2008 unterstützt jetzt die folgenden Datentypen:

- time
- date
- datetime2
- datetimeoffset

Für weitere Informationen zu diesen SQL-Servertypen siehe Date and Time Data Types and Functions (Transact-SQL).

Diese Art von Änderung führt dazu, dass der **`org.hibernate.test.hql.ASTParserLoadingTest`**-Test und der **`org.hibernate.test.stateless.StatelessSessionTest`**-Test fehlschlagen. Um das Problem zu umgehen, definieren Sie einen neuen Dialekt:

```
public class SQLServer2008Dialect extends SQLServerDialect
{
   public SQLServer2008Dialect()
   {
      registerColumnType( Types.DATE, "date" );
      registerColumnType( Types.TIME, "time" );
      registerColumnType( Types.TIMESTAMP, "datetime2" );
      registerFunction( "current_timestamp",
                        new NoArgSQLFunction("current_timestamp",
                                             Hibernate.TIMESTAMP,false) );
   }
}
```

JBPAPP-4731

Wurde ein Einfügungsvorgang mittels **`jconn3.jar`** ausgeführt, so wurden Null-Bit-Werte zu **`0`** konvertiert, damit keine Ausnahme gemeldet wurde. Sybase gestattet keinen Nullwert für den Bit-Datentyp, so dass Null-Bit-Werte zu Ausnahmen führen.

Um dieses Problem in Hibernate zu umgehen und den Wert wie beabsichtigt als Null zu persistieren, mappen Sie den Typ zu **`org.hibernate.test.where.NumericTrueFalseType`** statt **`boolean`**:

```
type="org.hibernate.test.where.NumericTrueFalseType"
```

JBPAPP-4334

Ein Fehler in jConnect bedeutet, dass die **`FetchingScrollableResultsImpl.isResultSetEmpty()`**-Methode folgendes wiedergibt, um zu bestimmen, ob ResultSet leer ist:

```
currentPosition == 0 && ! getResultSet().isBeforeFirst() && !
getResultSet().isAfterLast();
```

Ist das ResultSet leer, so sollte **`FetchingScrollableResultsImpl.isResultSetEmpty()`** "false" wiedergeben, Sybase JDBC jedoch "true". Dieses Problem lässt sich derzeit noch nicht umgehen.

JBPAPP-4175

Wenn Hibernate eine cachebare Anfrage unter Verwendung eines **`ResultTransformer`**

ausführt, so versucht es die Ergebnisse nach Anwendung des **`ResultTransformer`** zu cachen. Die Daten können jedoch verändert sein, so dass Hibernate sie nicht lesen kann. In diesem Fall wird eine **`ClassCastException`** beim Versuch die Ergebnisse zu cachen gemeldet.

JBPAPP-3913

Von Oracle 11g R2 (RAC) hat sich das Verhalten der CREATE TABLE Anweisung dahingehend geändert, als dass beim Erstellen einer konventionellen Tabelle in einer mit den Standardoptionen erstellten Datenbank das erste Segment nicht erstellt wird, ehe nicht die erste Reihe in die Tabelle eingefügt wurde (siehe http://download.oracle.com/docs/cd/E11882_01/server.112/e10595/tables002.htm#ADMIN13319 für Einzelheiten).

Als Folge dieser Änderung starten bei der ID-Generatorklasse **`sequence-identity`** als Standard erstellte Sequenzen mit 2 oder 4 statt mit 1. Zur Umgehung dieses Problems:

- Verwenden Sie eine andere ID-Generator Klasse oder
- Deaktivieren Sie die verschobene Segmenterstellung in Oracle durch Einstellung des Initialisierungsparameters **`DEFERRED_SEGMENT_CREATION`** auf **`FALSE`**. Die CREATE TABLE Anweisung hat zwei neue Klauseln (SEGMENT CREATION DEFERRED und SEGMENT CREATION IMMEDIATE), die den Wert des **`DEFERRED_SEGMENT_CREATION`**-Parameters außer Kraft setzen.

JBPAPP-3911

In Oracle 11g R2 (sowohl RAC als auch Standalone) kann eine komplexe Anfrage mit **`LockMode.UPGRADE`** (d.h. "for update") einen "No more data to read from socket"-Fehler verursachen. Dies ist kein Hibernate-Bug, sondern hat mit der Datenbankkonfiguration zu tun. Weitere Informationen finden Sie unter Bug Hunting. Sie umgehen das Problem, indem Sie **`LockMode.UPGRADE`** bei solchen Anfragen vermeiden.

JBPAPP-3483

Hibernate protokolliert derzeit nicht die Ausführungszeit jeder Anfrage. Dies wird sich wahrscheinlich in zukünftigen Versionen der JBoss Enterprise Application Platform ändern.

JBPAPP-3284

Dies ist ein Problem mit cglib, welches in dieser Release der JBoss Enterprise Application Platform veraltet ist. Weitere Einzelheiten finden Sie unter **`http://sourceforge.net/tracker/index.php?func=detail&aid=2796998&group_id=56933&atid=482368`**. cglib entfernt Feldannotationen bei der Transformation einer Klasse mit **`TransformingClassGenerator`** wegen eines Problems mit der **`visitField`**-Methode. Sie können dieses Problem umgehen, indem Sie "Getter" statt Felder annotieren, allerdings ist dies eventuell nicht für schwere Applikationen möglich, die zu JBoss Enterprise Application Platform 5 portiert werden.

JBPAPP-3223

Hibernate unterstützt derzeit keine Tuple-Syntax in HQL oder Criteria in Datenbanken, die Tuple-Syntax nicht unterstützen. Wenn eine Datenbank zum Beispiel Tuple-Syntax nicht unterstützt:

```
wo (a,b) in ( (1,2), (3,4) )
```

Sollte Hibernate übersetzen:

```
wo ( (a=1 UND b=2) ODER ( (a=3 UND b=4) )
```

Derzeit gibt es keine Möglichkeit, dieses Problem zu umgehen, außer diese Syntaxanfrage bei Datenbanken, die die Tuple-Syntax nicht unterstützen, zu vermeiden.

JBPAPP-3075

Wird ein für die Datenbank reservierter Schlüssel als Property-Name mit einer Hibernate Validator Annotation verwendet (zum Beispiel **`@Min`** oder **`@Max`**), so werden Ausnahmen in **`SchemaExport`** gemeldet, selbst wenn Sie einen Spaltennamen festlegen. Der Grund hierfür ist, dass Hibernate den festgelegten Namen ignoriert. Sie umgehen das Problem, indem Sie den Property-Namen an etwas mappen, das kein für die Datenbank reservierter Schlüssel mit der **`@Column`**-Annotation ist. Die Behebung des Problems wird in Hibernate 4 erwartet.

JBPAPP-3069

Der **`QueryByExampleTest.testJunctionNotExpressionQBE`**-Test schlägt in Sybase fehl, weil ansinull standardmäßig deaktiviert ist. Dieser Test erstellt ein ODER-Verknüpfungsprädikat wie **`( OR^ (ex) (NOT ex) )`**. Dies sollte mit allem in der Datenbank übereinstimmen, da aber ANSI SQL alle NULL-Werte involvierenden Vergleiche als UNKNOWN evaluiert, wurden nicht alle Übereinstimmungen wiedergegeben. Um dieses Problem zu umgehen, hängen Sie folgenden String an die JDBC-URL an:

```
?SQLINITSTRING=set ansinull on
```

Ist dies nicht möglich, so besteht eine Alternative darin, den folgenden Java-Code (oder einen ähnlichen) nach Erhalt einer Hibernate Session **`s`** auszuführen:

```
s.connection().createStatement().execute("set ansinull on");
```

JBPAPP-3034

Beim Ordnen von Batch-Insert-Anweisungen werden eingebettete Klassen nicht berücksichtigt. Es gibt zwei Möglichkeiten, wie dieses Problem umgangen werden kann. Setzen Sie für die erste Möglichkeit **`ORDER_INSERTS`** auf **`FALSE`**, wenn eingebettete Klassen verwendet werden. Die zweite Möglichkeit ist, explizit **`session.save()`** auf untergeordneten Objekten aufzurufen, um deren SQL Insertion Orders zu erzwingen.

JBPAPP-3032

MySQL unterstützt derzeit keine Millisekunden- und Mikrosekunden-Einheiten bei der Rückgabe von Datenbankwerten wie **`TIME`** und **`TIMESTAMP`**.

JBPAPP-3031

Auf Sybase wird der **`current_timestamp`**-Text vom Übersetzer nicht als ein *Methodenmodus* erkannt. Es gibt derzeit keine andere Möglichkeit, dieses Problem zu umgehen, als zu vermeiden, auf die Funktionsersetzung für **`current_timestamp`** angewiesen zu sein.

JBPAPP-3030

Auf Sybase kann **`SchemaExport`** nicht dazu verwendet werden, im verketteten Transaktionsmodus gespeicherte Prozeduren zu erzeugen. Fügen Sie zur Umgehung dieses Problems den folgenden Code unmittelbar nach der Definition der neuen gespeicherten Prozedur ein:

```
<database-object>
      <create>
      sp_procxmode paramHandling, 'chained'
      </create>
      <drop/>
      </database-object>
```

JBPAPP-3010

Die **`evict(Object)`**-Methode in **`EntityRegionAccessStrategy`** und **`CollectionRegionAccessStrategy`** versucht, Objekte aus dem Cache ohne Rücksicht auf die Transaktionsisolation zu entfernen. Dies wird derzeit nicht unterstützt, da die **`removeNode`**-Methode von JBoss Cache nicht mit Transaktionen arbeitet.

JBPAPP-2945

Das Einstellen einer Anfragezeitüberschreitung für ein PreparedStatement wird von PostgreSQL 8.3.7 nicht unterstützt. Diese Einschränkung bedeutet, dass Anfragen scheitern werden, wenn sie eine Annotation wie die folgende verwenden:

```
        @QueryHint(name = "org.hibernate.timeout", value = "100")
```

JBPAPP-2971

Die Implementierung von Applikationen, die auf Hibernate zur Verwendung von **`cglib`** als Byte-Provider verweisen, schlägt aufgrund einer **`java.lang.SecurityException`** fehl. Eine Fehlermeldung ähnlich der folgenden wird ausgegeben:

```
Deployment "persistence.unit:unitName=lobtest.ear/
      lobtest-ejb-1.0-SNAPSHOT.jar#lobtest-jpa-jndi" is in error due to the
following reason(s):
      java.lang.SecurityException: class
      "com.redhat.gss.lobtest.jpa.Item$$EnhancerByCGLIB$$defd1a7f"'s signer
information does not
      match signer information of other classes in the same package
```

Der Grund hierfür liegt darin, dass **`cglib.jar`** in JBoss Enterprise Application Platform 5.0 signiert ist, und der mit cglib instrumentierte Proxy die **`cglib.jar`**-Signaturinformationen nutzt anstelle der Signaturinformationen der Zielklasse der Applikation.

cglib ist in der 5.1 Release der JBoss Enterprise Application Platform veraltet.

JBPAPP-2867

Sybase unterstützt derzeit keine Hibernate **`Blob`**s oder **`Clob`**s und Hibernate unterstützt keine Sybase **`text`** oder **`image`** Datentypen. Um dieses Problem zu umgehen, erzeugen Sie benutzerdefinierte Typen, die auf die Sybase **`text`** und **`image`** Typen verweisen.

JBPAPP-2792

Sybase fügt keine neue Entity ein, wenn sie ihre Spalte überfüllt. Allerdings meldet es keine Ausnahme, so dass Hibernate nicht bemerkt, dass das Einfügen fehlschlug. Um dieses Problem zu umgehen, verwenden Sie das **`jconn3.jar`** mit **`DYNAMIC_PREPARE=true`** in der Hibernate Konfigurationsdatei.

```
<property name="connection.url">jdbc:sybase:Tds:aurum:1503/masterDb?
  DYNAMIC_PREPARE=true</property>
```

Eine andere Umgehung des Problems ist die Verwendung von **`jconn4.jar`**.

JBPAPP-2789

`ShemaExport` schlägt auf Oracle und Sybase fehl, wenn eine redundante **`@Column(unique=true)`** oder **`UniqueContraint(columnnames={...})`** Annotation in einer Spalte verwendet wird, die vom deklarierten Modell implizit als eindeutig definiert ist. Um dieses Problem zu umgehen, entfernen Sie die redundante **`@Column(unique=true)`** oder **`UniqueContraint(columnnames={...})`** Annotation.

JBPAPP-2613

Wenn der DB2 v9.7 Treiber mit progressivem Streaming verwendet wird (dies ist der Standard), schlagen Operationen auf **`Blob`** und **`Clob`**-Locators fehl. Um dieses Problem zu umgehen, gibt es zwei Möglichkeiten:

- Erstellen Sie eine Property-Datei namens **`DB2JccConfiguration.properties`** entweder in Ihrem Klassenpfad oder in einem JAR in Ihrem Klassenpfad. Fügen Sie dieser Datei folgende Zeile hinzu, um progressives Streaming in DB2 zu deaktivieren:

  ```
  db2.jcc.override.progressiveStreaming=2
  ```

- Verwenden Sie den DB2 Version 9.1 Treiber statt DB2 Version 9.7.

JBPAPP-2408

Es besteht ein Problem mit dem DB2 v9.7 Treiber, wenn eine Identität oder ein nativer ID-Generator mit Hibernate verwendet wird. Die **`Statement.getGeneratedKeys()`**-Treibermethode in DB2 gibt eine leere *Ergebnistabelle* wieder anstelle der generierten Schlüssel. Infolgedessen meldet Hibernate eine Ausnahme, die besagt, dass die Datenbank keinen nativ generierten Identitätswert wiedergab. Dieses Problem wurde in der Version des DB2 9.7 JDBC Treibers behoben, der mit Data Studio 2.2 bereitgestellt wird, und steht auf der DB2-Website zum Download zur Verfügung. Diese Version wird zum Einsatz mit Hibernate empfohlen.

JBPAPP-2278

Die **`Save`** (speichern) Operation kann fehlschlagen, wenn eine flüchtige Entity über mehrere Pfade erreicht werden kann und mindestens einer dieser Pfade nicht für die **`Save`**-Operation kaskadiert. Um dieses Problem zu umgehen, sollten Sie die flüchtige Entity zunächst speichern, bevor Sie die zuvor fehlgeschlagene save-Operation ausführen. Ist dies nicht möglich, können Sie alternativ entweder einen oder beide Cascade- und Entity-Mappings ändern, um die Reihenfolge der Cascade-Pfade derart zu ändern, dass die flüchtige Entity gespeichert wird,

bevor zu der Entity kaskadiert wird, die eine nicht-flüchtige Entity erfordert.

JBPAPP-1895

Die Hibernate Tests **`JoinTest.java`** und **`QueryAndSQLTest.java`** schlagen fehl, wenn Anfragen mit typenlosen Parametern für DB2 getestet werden, welches typenlose Parameter nicht unterstützt. Sie können dieses Problem umgehen, indem Sie die Anfragen so bearbeiten, dass die Parameter an die passenden Datentypen verteilt werden wie im JIRA beschrieben.

JBPAPP-1613

Nullwerte für Spalten markiert als **`Boolean`** in Sybase werden als **`0`** anstelle von **`null`** persistiert. Weisen Sie **`type="org.hibernate.test.where.NumericTrueFalseType"`** anstelle von **`type="boolean"`** zu, um dieses Problem zu umgehen.

JBPAPP-1554

Sybase erlaubt nur einen Eintrag (z.B. Spaltenname oder '*') in einer Unteranfrage-Select-Liste. Die HQL-Funktion, **`elements()`**, schlägt fehl, wenn das Collection-Element eine zusammengesetzte ID besitzt, da das generierte SQL eine Unteranfrage-Select-Liste mit mehreren Einträgen enthält. Um das Problem zu umgehen, vermeiden Sie die Verwendung von HQL **`elements()`**, wenn die Elemente einen zusammengesetzten Schlüssel enthalten. Wandeln Sie stattdessen die HQL so um, dass keine Unterabfrage mehrere Einträge in ihrer Auswahlliste hat.

JBPAPP-1545

In Sybase, wenn eine Abfrage ein ANSI-Join besitzt mit drei oder mehr Joins und einer dieser Joins eine Union-Unterklasse umfasst, kann die Abfrage mit einer **`SybSQLException`** fehlschlagen, da eine Spalte nicht innerhalb des Geltungsbereichs des Joined-Table-Ausdrucks liegt. Derzeit wird empfohlen, die Verwendung von Join-Fetches im Zusammenhang mit Union-Unterklassen zu vermeiden.

JBPAPP-1230

Wenn eine **`DetachedCriteria`** als Unterabfrage verwendet wird, enthält die generierte SQL einen Spaltenalias in der Unterabfrage. Auf Sybase wird dadurch eine **`SybSQLException`** gemeldet, da Sybase keine Spaltenaliasse in Unterabfragen zulässt. Um dieses Problem zu umgehen, verwenden Sie in einer Unterabfrageeine eine HQL-Abfrage anstelle einer **`DetachedCriteria`**.

JBPAPP-1123

Wenn **`@OrderBy`** auf Joined-Klassen verwendet wird (mittels einer Join-Tabelle), ist das generierte SQL auf MySQL, PostgreSQL, Oracle und MSSQL ungültig, da die "order by"-Klausel die Spalten unter Verwendung des tatsächlichen Tabellennamens qualifiziert. Die "order by"-Klausel sollte stattdessen den Tabellenalias verwenden.

JBPAPP-1082

Wenn eine **`char`**-Eigenschaft verwendet wird und diese nicht initialisiert wird, so initialisiert Hibernate sie zu **`0`** und persistiert einen String, der das Zeichen **`\u0000`** enthält. PostgreSQL meldet eine Ausnahme, da es das Zeichen **`\u0000`** nicht in einem String eingebettet zulässt. Das Problem der Persistierung von **`\u0000`** in einer **`char`**-Spalte mittels PostgreSQL lässt sich

derzeit nicht beheben.

Falls der Benutzer beabsichtigt, eine NULL als eine nicht initialisierte **`char`**-Property anstelle von **`\u0000`** zu persistieren, dann sollte **`java.lang.Character`** verwendet werden anstelle des primitiven **`char`**-Typs. Dies verhindert die Ausnahme beim Initialisieren der Property. Der Versuch, eine **`java.lang.Character`**-Property zu persistieren, die auf **`\u0000`** gesetzt ist, verursacht nach wie vor eine Ausnahme.

JBPAPP-1071

In manchen Fällen, wenn Fremdschlüsseleinschränkungen auf Spalten in einem Primärschlüssel definiert sind, deklariert **`SchemaExport`** diese fälschlicherweise als nullbar, wenn es **`CREATE TABLE`**-Anweisungen generiert. Dies schlägt auf MSSQL, DB2 und Sybase fehl, da diese Datenbanken erfordern, dass Primärschlüsselspalten nicht nullbar sind.

Um dieses Problem zu umgehen, legen Sie eindeutig fest, welche Spalten nicht nullbar sein sollen, z.B:

- Fügen Sie **`nullable=false`** zu **`@JoinColumn`** hinzu
- Fügen Sie **`optional=false`** zu **`@ManyToOne`** hinzu
- Fügen Sie **`@AttributeOverride`** mit **`@Column(name="mapkey", nullable=false)`** hinzu, falls eine **`@CollectionOfElements`** eine Map verwendet
- Fügen Sie **`nullable=false`** in **`@Column`** hinzu, wenn innerhalb einer **`@CollectionId`** oder innerhalb einer **`@MapKey`**

10.5. Bekannte Probleme mit JBoss Messaging

JBPAPP-5124

Bei der Verwendung der Sybase-Datenbank mit dem JDBC-Treiber Sybase jConnect JDBC driver v7 (Build 26502), kann ein **`MaxParams`**-Attribut in der **`sybase-persistence-service.xml`**-Konfigurationsdatei nicht höher als **`481`** eingestellt werden, da es eine Einschränkung der **`PreparedStatement`**-Klasse in diesem Treiber gibt. Unerwartete Fehlschläge können vorkommen, wenn das **`MaxParams`**-Attribut auf einen höheren Wert als **`481`** eingestellt ist. Um dieses Problem zu umgehen, stellen Sie **`MaxParams`** auf einen Wert ein, der kleiner als **`481`** ist.

JBPAPP-4668 (ehemals JBPAPP-3965)

Zwei JBoss Messaging Test Suite Tests schlagen bei Oracle 11g R1, R2 und RAC mit dem aktuellsten JDBC-Treiber, Version 11.2.0.1.0 fehl:

- QueueManagementTest.testDestroyDestinationProgrammatically
- QueueManagementTest.testDestroyDestinationProgrammaticallyWithParams

Diese Tests einen großen Wert als den fullSize-Warteschlangenkonfigurationsparameter, der an die **`setFetchSize`**-Method am **`java.sql.PreparedStatement`** gegeben wird. Ein Problem mit dem JDBC-Treiber bedeutet, dass mehr als die übliche Menge an Speicher verbraucht wird wenn **`executeQuery()`** aufgerufen wird, was zu einem **`java.lang.OutOfMemoryError`** führt, wodurch der Test fehlschlägt.

JBPAPP-3904

Oracle JDBC Treiberversion 11.1.0.7.0 führt dazu, dass die JBoss Messaging Test Suite bei Oracle 11g R1 mit einer **`SQLException`** ("Bigger type length than Maximum") fehlschlägt. Der Grund hierfür ist eine Regression im Oracle JDBC Treiber 11.1.0.7.0. Wir empfehlen Oracle JDBC Treiberversion 11.2.0.1.0 zum Gebrauch mit Oracle 11g R1, Oracle 11g R2, Oracle RAC 11g R1 und Oracle RAC 11g R2.

JBPAPP-3352

Wird ein message-driven Bean mit den Standardeinstellungen deployt ((useDLQ=true, DLQMaxResent=5) und die erneute Nachrichtenlieferung wird aufgefordert, so verbleiben Nachrichten in der Warteschlange in einem "Liefer"-Status, selbst nachdem die Nachricht an die "dead letter"-Warteschlange geliefert wurde. Sie umgehen dieses Problem, indem Sie die Verwendung der "dead letter queue" deaktivieren: useDLQ=false.

JBPAPP-2033

`EnableMessageCounters` auf der **`jboss.messaging.ServerPeer`** JMX-Schnittstelle kann nicht auf **`true`** gesetzt werden. Um dennoch die Nachrichtenzähler zu aktivieren, rufen Sie auf derselben JMX-Schnittstelle die **`enableMessageCounters()`**-Operation auf.

10.6. Bekannte Probleme mit JBoss Web Services

JBPAPP-5021

Die **`jbossws-cxf-installer/docs/Install.txt`**-Datei im JBoss WS CXF Installer nennt JBoss WS CXF fälschlicherweise eine Technologievorschau. Der Text von **`Install.txt`** sollte wie folgt lauten:

```
Führen Sie einfach "ant" vom jbossws-cxf-installer Verzeichnis aus, das vom
jbossws-cxf-installer.zip in Ihr JBOSS_HOME für Ihre JBoss Enterprise
Platform extrahiert wurde.
```

JBPAPP-4561

Bei JBoss Web Services Native wird, wenn MTOM aktiviert ist und die DataHandler "getter"-Signatur einer Return-Type Klasse nicht mit der JavaBean "getter"-Signature übereinstimmt, wird die Antwort als ein Inline-String gesendet statt dass XOP verwendet wird. Dies ist der Fall, weil das Scannen nach Attachment-Annotationen auf der JavaBean "getter"-Signatur basiert.

JBPAPP-4506

Wenn Dienste an JBoss Web Services Native deployt werden, werden jegliche XML Schema Declaration (XSD) Importe mit relativen URLs relativ zum WSDL-Dienstvertrag als absolute URLs umgeschrieben. Importiert eine importierte XSD eine zweite XSD unter Verwendung eines relativen Pfads, so wird dieser Pfad fälschlicherweise dahingehen umgeschrieben, dass er relativ zur XSD statt dem WSDL-Dienstvertrag ist. Sie umgehen dieses Problem, indem Sie alle XSD- und WSDL-Dateien in dasselbe Verzeichnis verschieben und alle Importe ändern.

JBPAPP-3028

Die Aktualisierung von JBoss Web Services 2.0.1.SP zu JBoss Web Services Native 3.1.2.SP bringt viele Änderungen und neue Funktionen ein. Die zusätzlich nötige Verarbeitungszeit für mehrere neue Funktionen (Ressourceninjektion, Unterstützung für **`@PostConstruct`** und **`@PreDestroy`**, etc.) verursacht eine leichte Beeinträchtigung der Gesamtleistung.

10.7. Bekannte Probleme mit Remoting

JBPAPP-3392

EJB3 Clients verwenden bestehende Socket Connections bei nachfolgenden Aufrufen nicht; eine neue Verbindung wird pro Aufruf erstellt.

10.8. Bekannte Probleme mit RESTEasy

JBPAPP-5038

Die **`jettison.jar`**-Datei ist nicht im **`jboss-eap-5.1/resteasy/lib`**-Verzeichnis von **`jboss-eap-noauth-5.1.0.zip`** enthalten. Um dieses Problem zu umgehen, verwenden Sie die **`jettison.jar`**-Datei, die in der Seam Distribution enthalten ist, **`jboss-eap-5.1/seam/lib/jettison.jar`**.

JBPAPP-2993

Das **`spring-hibernate-contacts`**-Beispiel schlägt mit dem Fehler **`java.lang.IllegalArgumentException: object is not an instance of declaring class`** fehl. Dies ist der Fall weil der **`getContactById(@PathParam("id") Long id)`**-Methode in der **`ContactServiceImpl`**-Klasse die @GET-Annotation fehlt.

10.9. Bekannte Probleme mit Seam

JBPAPP-4808

Von POJO Cache mit Session-Replikation auf Feldebene gesicherte Seam Applikationen können mit einer **`InstantiationException`** fehlschlagen. Der Grund hierfür liegt in der Tatsache, dass die Architektur des POJO Cache spezifische Funktionalitäten und Verhalten von Clients erfordert, um Session-Replikation auf Feldebene zu bieten.

Es existiert eine Problemumgehung. Fügen Sie der JSF-Konfiguration den folgenden Code (**`faces-config.xml`**) hinzu

```
<context-param>
  <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
  <param-value>client</param-value>
 </context-param>>
```

JBPAPP-4508

Beim Hochladen von Dateien mit <s:fileUpload> in <h:dataTable>, treten Fehler auf, die zu identischen Dateinamen und inkorrekten Inhalten führen. Dies ist der Fall, weil der lokale Wert nicht von der **`getLocalValue()`**-Methode wiedergegeben wird.

JBPAPP-4231

Eine **`NullPointerException`** wird gemeldet, wenn Seam-Beispiele undeployt ode deinstalliert werden. Dies führt nicht zum Herunterfahren der Applikation, jedoch sollte dieser Fehler nicht vorkommen. Dieses Problem wurde in den nestedbooking, dvdstore, itext und excel Beispiel beobachtet, tritt aber im ui-Beispiel nicht auf.

JBPAPP-3546

Das iText-Beispiel enthält einen "Programming Skills"-Abschnitt. In der Auswahlliste gewählte Posten sollten in der generierten PDF enthalten sein. Aufgrund eines Fehlers in iText wird bei der Auswahl mehrerer Posten nur der erste in die generierte PDF aufgenommen.

JBPAPP-2385

Seam's Spring-Beispiel schlägt mit einer **`IllegalStateException`** fehl, wenn das Anmeldeformular auf der IBM Virtual Machine abgeschickt wird. Dies liegt an einem Defekt in der IBM Virtual Machine. Dieses Problem wird nicht behoben, bis nicht zunächst der Fehler in der IBM Virtual Machine beseitigt wird.

JBPAPP-2377

Das Seamspace-Beispiel schlägt mit einer **`NullPointerException`** fehl, wenn ein neuer Blog-Eintrag auf der IBM Virtual Machine abgeschickt wird. Dies liegt an einem Defekt in der IBM Virtual Machine. Dieses Problem wird nicht behoben, ehe nicht zunächst der Fehler in der IBM Virtual Machine beseitigt wird.

10.10. Bekannte Probleme mit EJB3

JBPAPP-4308

Die folgenden Klassen wurden dem jboss-ejb3-deployer.jar hinzugefügt:

```
        org/jboss/ejb3/iiop/
        org/jboss/ejb3/iiop/BeanCorbaServant.class
        org/jboss/ejb3/iiop/EJB3IIOPWebClassLoader.class
        org/jboss/ejb3/iiop/IORFactory$1.class
        org/jboss/ejb3/iiop/IORFactory$2.class
        org/jboss/ejb3/iiop/IORFactory.class
        org/jboss/as/ejb3/deployers/IORDeployer.class
```

Die @Remote* Annotationen in der Enterprise Application Platform 'web' Konfiguration und Enterprise Web Platform melden eine NullPointerException. Dies ist der Fall, weil **`jboss-iiop.jar`** nicht in der Webkonfiguration inbegriffen ist, die im Wesentlichen die Enterprise Web Platform ist.

JBPAPP-4692

Das Deployment einer Persistenzeinheit in ein EAR schlägt fehl, wenn die Einheit sich außerhalb des EAR befindet und das Bean, das die Persistenzeinheit einzuspeisen versucht sich innerhalb des EAR befindet. Die Einspeisung schlägt fehl, weil die Persistenzeinheit nicht aufgefunden werden kann. Dies ist zu erwartendes Verhalten für die EJB3 Spec. Für strikte EJB3-spec Konformität sollte die Persistenzeinheit innerhalb des EARs verpackt sein.

Ein JBoss-spezifisches Verhalten gestattet Persistenzeinheiten die Existenz außerhalb von EARs. Dies ist in der **`deployers/ejb3.deployer/META-INF/jpa-deployer-jboss-beans.xml`**-Datei unter einem JBoss AS Serverprofil konfiguriert. Der relevante Abschnitt lautet:

```
<!--
     Can be DefaultPersistenceUnitDependencyResolver for spec compliant
resolving,
     InterApplicationPersistenceUnitDependencyResolver for resolving beyond
EARs,
     or DynamicPersistencePersistenceUnitDependencyResolver which allows
configuration via JMX.
  -->
  <bean name="PersistenceUnitDependencyResolver"
class="org.jboss.jpa.resolvers.DynamicPersistenceUnitDependencyResolver"/>>
```

Standardmäßig wird der **`DynamicPersistenceUnitDependencyResolver`** (https://svn.jboss.org/repos/jbossas/projects/jpa/trunk/deployers/src/main/java/org/jboss/jpa/resolvers/DynamicPersistenceUnitDependencyResolver.java) verwendet, der die Steuerung des spec-konformen Verhaltens mittels MBean in der JMX-Konsole gestattet. Die spec-nicht-konforme JBoss Varianten Suchstratehie finden Sie unter https://svn.jboss.org/repos/jbossas/projects/jpa/trunk/deployers/src/main/java/org/jboss/jpa/resolvers/strategy/JBossSearchStrategy.java.

10.11. Bekannte Probleme mit Clustering

JBPAPP-3795

Wird ein aufgeklapptes Archiv deployt, das eine 0-byte Datei (neben anderen Dateien) enthält, so wird eine NullPointerException am Remote-Node gemeldet. Die bricht den Farm Deployment Vorgang ab. Das Problem lässt sich derzeit nicht umgehen.

10.12. Bekannte Probleme mit dem Connector

JBPAPP-5048

Die Statusseite des mod_cluster Manager-Moduls wird beim Worker-Node-Failover nicht aktualisiert. Das bedeutet, dass der Worker als aktiv und verfügbar gelistet ist, nachdem er fehlgeschlagen ist.

JBPAPP-4767

Es kommt zu folgenden Fehlerprotokollmeldungen wenn der Client unerwartet die Verbindung in AJP schließt:

```
              [error] [client <ip>] proxy: error processing body
              [error] proxy: dialog to 192.168.0.1:8009 (192.168.0.1)
failed
```

Diese Fehler sollten als Warnung und Fehlerbehebung statt als Fehler protokolliert werden.

JBPAPP-3463

Wenn eine Applikation undeployt wird, werden Sessions ungültig, die vom mod_cluster an den Applikationsserver geleitet wurden ehe die undeploy-Benachrichtigung erhalten wurde, was den Fehler "503 - This application is not currently available" ("503 - Diese Applikation ist derzeit nicht erreichbar") zur Folge hat.

MODCLUSTER-123

Wird der **`root`**-Kontext ("`/`") deployt und aktiviert, so können andere Kontexte nicht deaktiviert werden. Es ist außerdem nicht möglich festzulegen, dass andere Kontexte nicht an den **`root`**-Kontext weitergeleitet werden.

A. Änderungsverzeichnis

Version 5.1.0-2.400	**2013-10-31**	**Rüdiger Landmann**

Rebuild with publican 4.0.0

Version 5.1.0-2	**2012-07-18**	**Anthony Towns**

Rebuild for Publican 3.0

Version 5.1-0	**Fri Sep 03 2010**	**Laura Bailey**

Entwurf der Release Notes für EAP 5.1 GA.